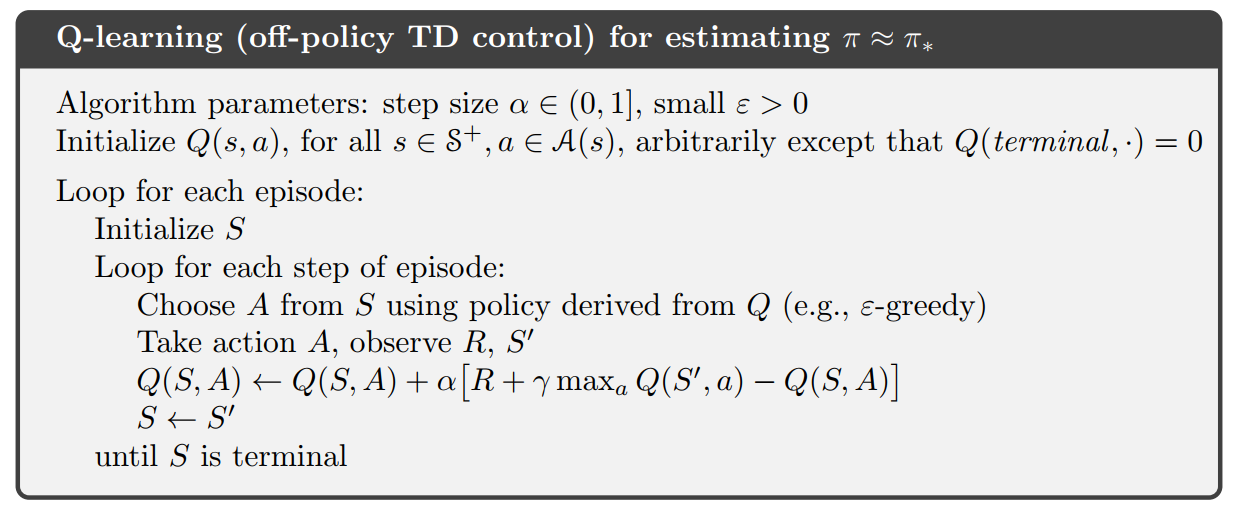
**My First Reinforcement Learning**

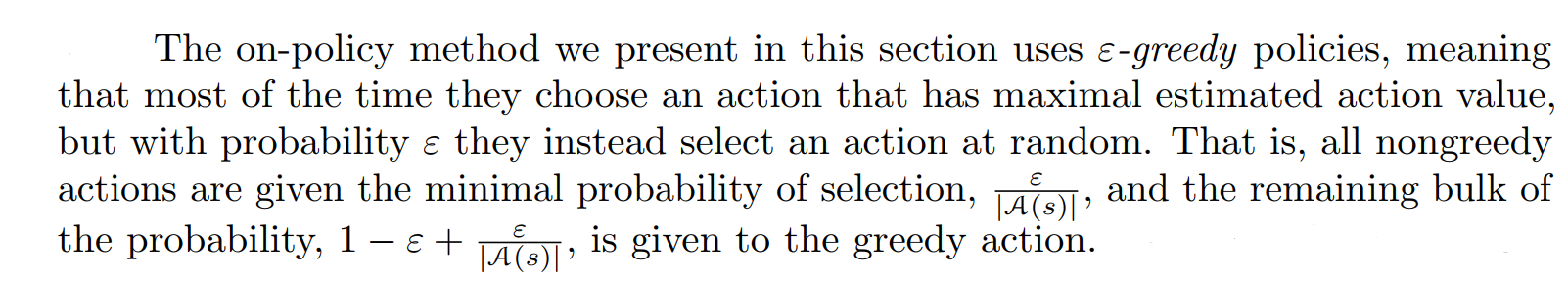
* **Number of States:** 70 (7 x 10 grid)
* **Number of Actions:** 4 [Up (0), Right (1), Down (2), Left (3)]

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:

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.



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

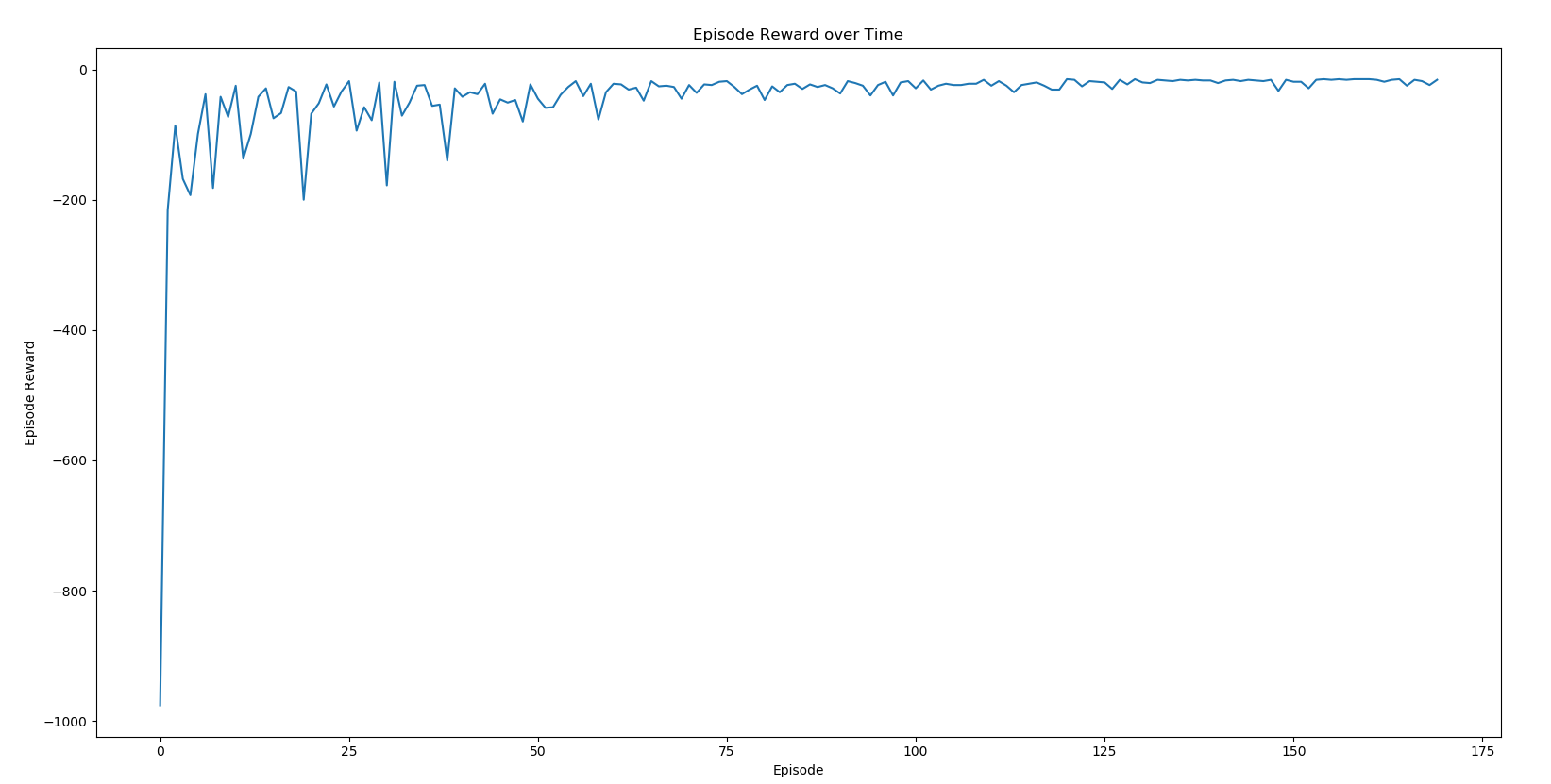


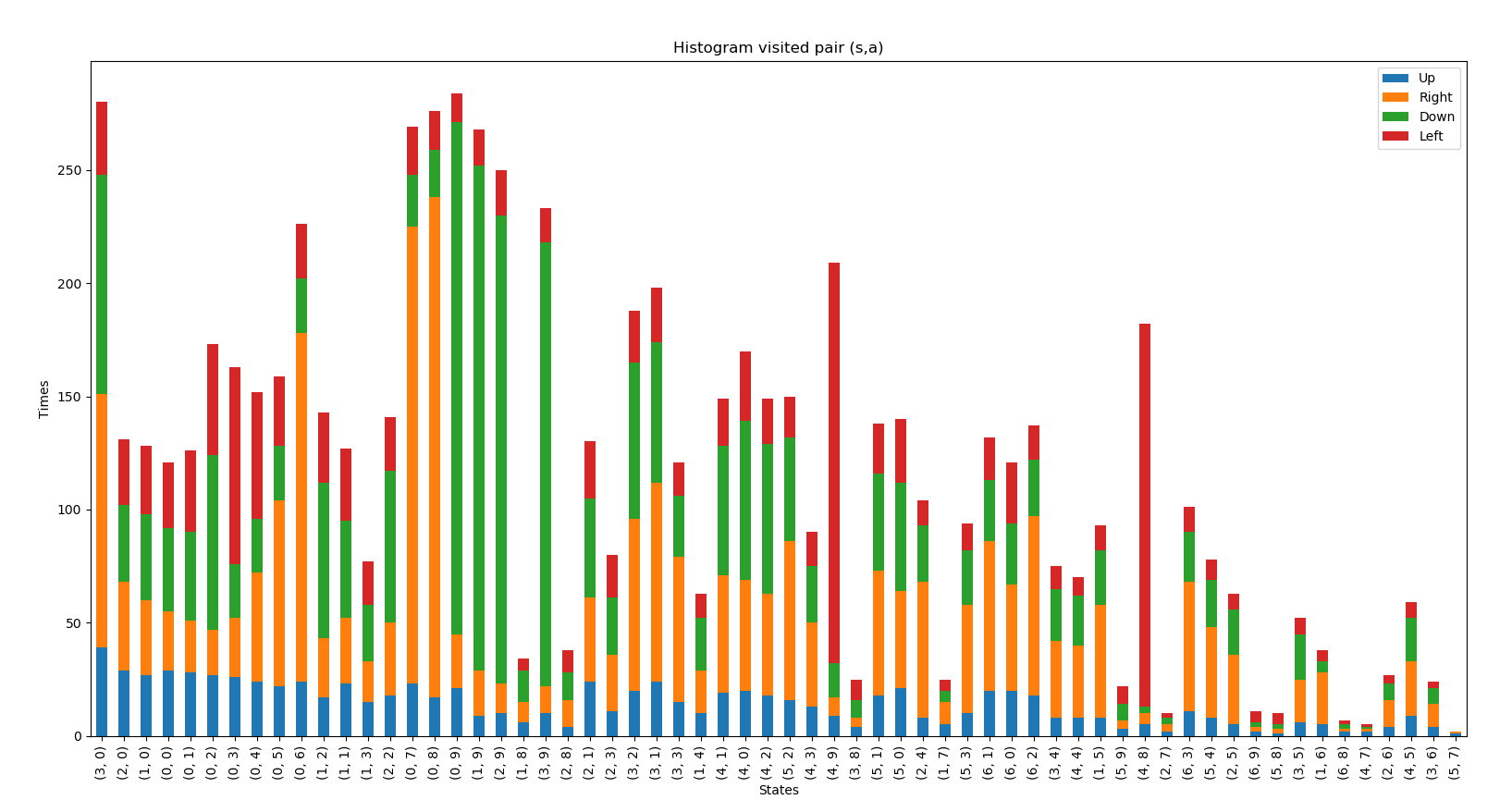
The states and actions of the environment are given by the Gym library as well as the reward in every state and a “done” flag when the agent steps in the goal state. The Q-Values are stored inside a dictionary in python and are initialized in 0, in other words, every state is initialized with the optimal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

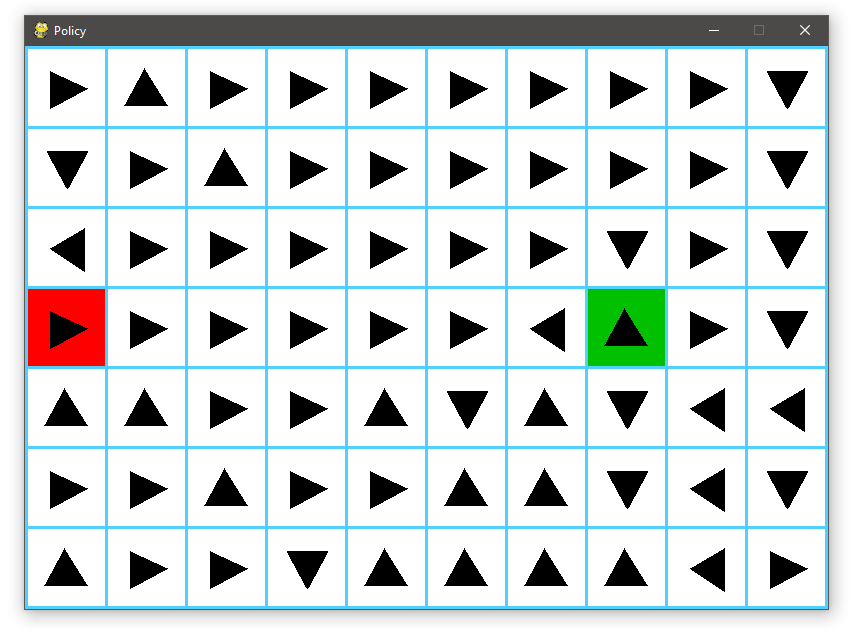
The firs try made with the implemented algorithm we used the same values for alpha, gamma, and epsilon provided by the solution given in the book:

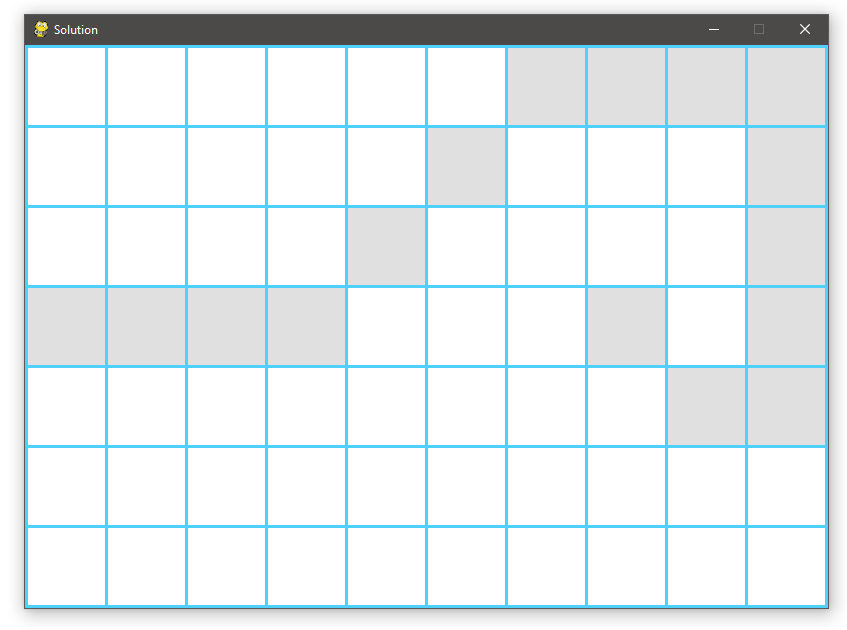
* **Alpha (α):** 0.5
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 170

The obtained results are shown below:

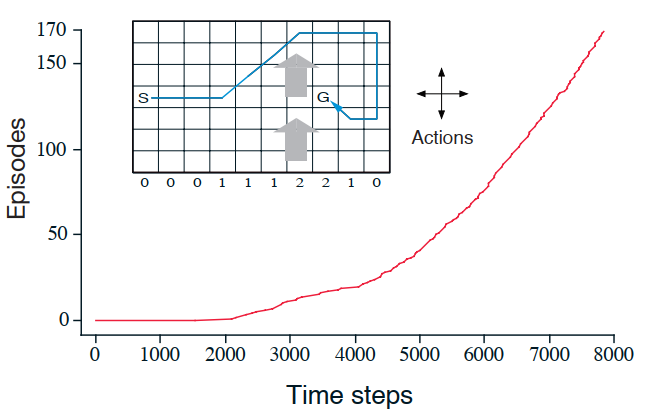








And as seen, the final result is completely satisfactory, fulfilling the solution given by the book.



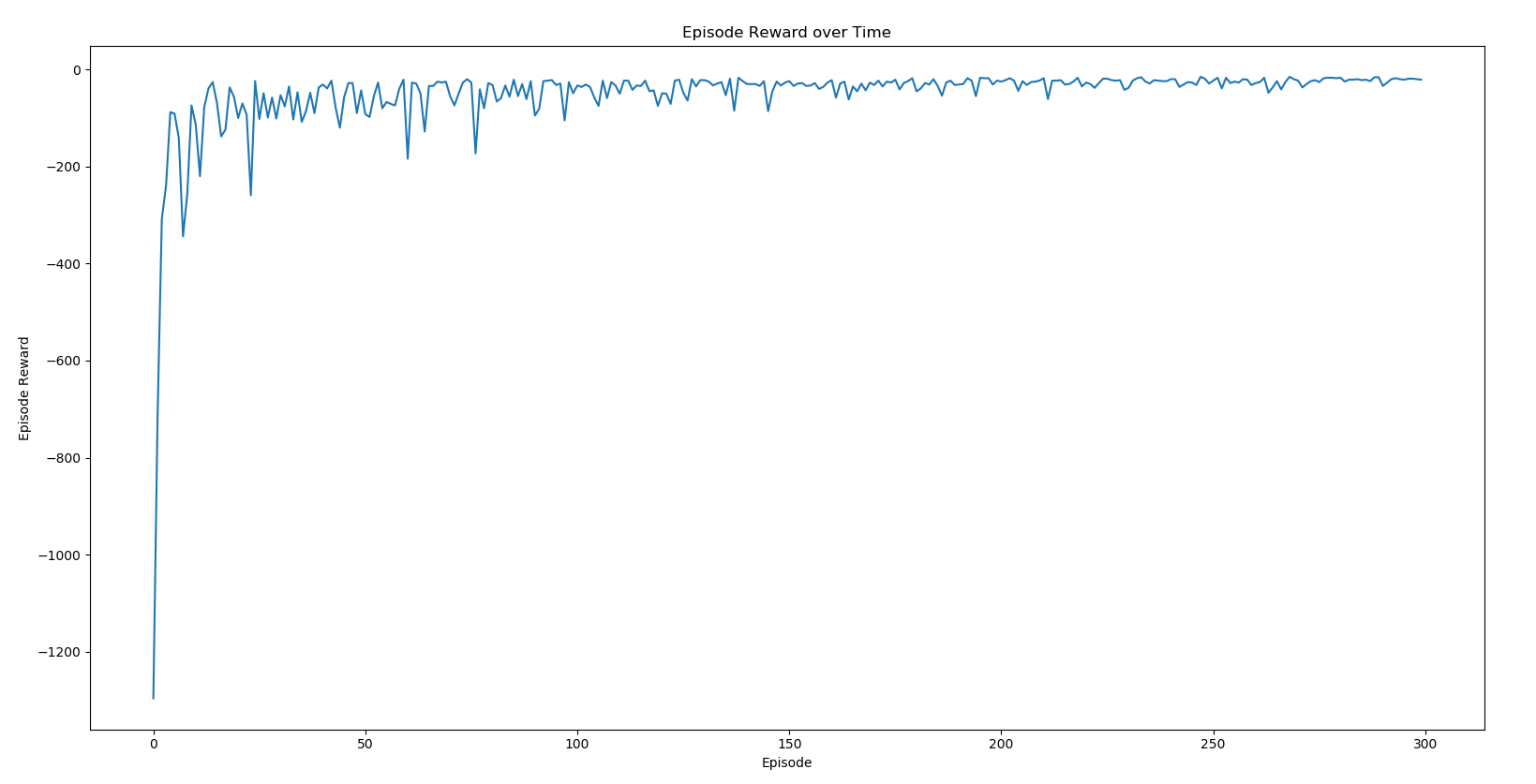
To realize the hyper-parameter sweeping, we decided to sweep 3 times each variable, and in the case of alpha, gamma and epsilon to implement a little piece of code in which the algorithm will use the minimal quantity of episodes to resolve the gridworld, this with the sake of comparison between variations.

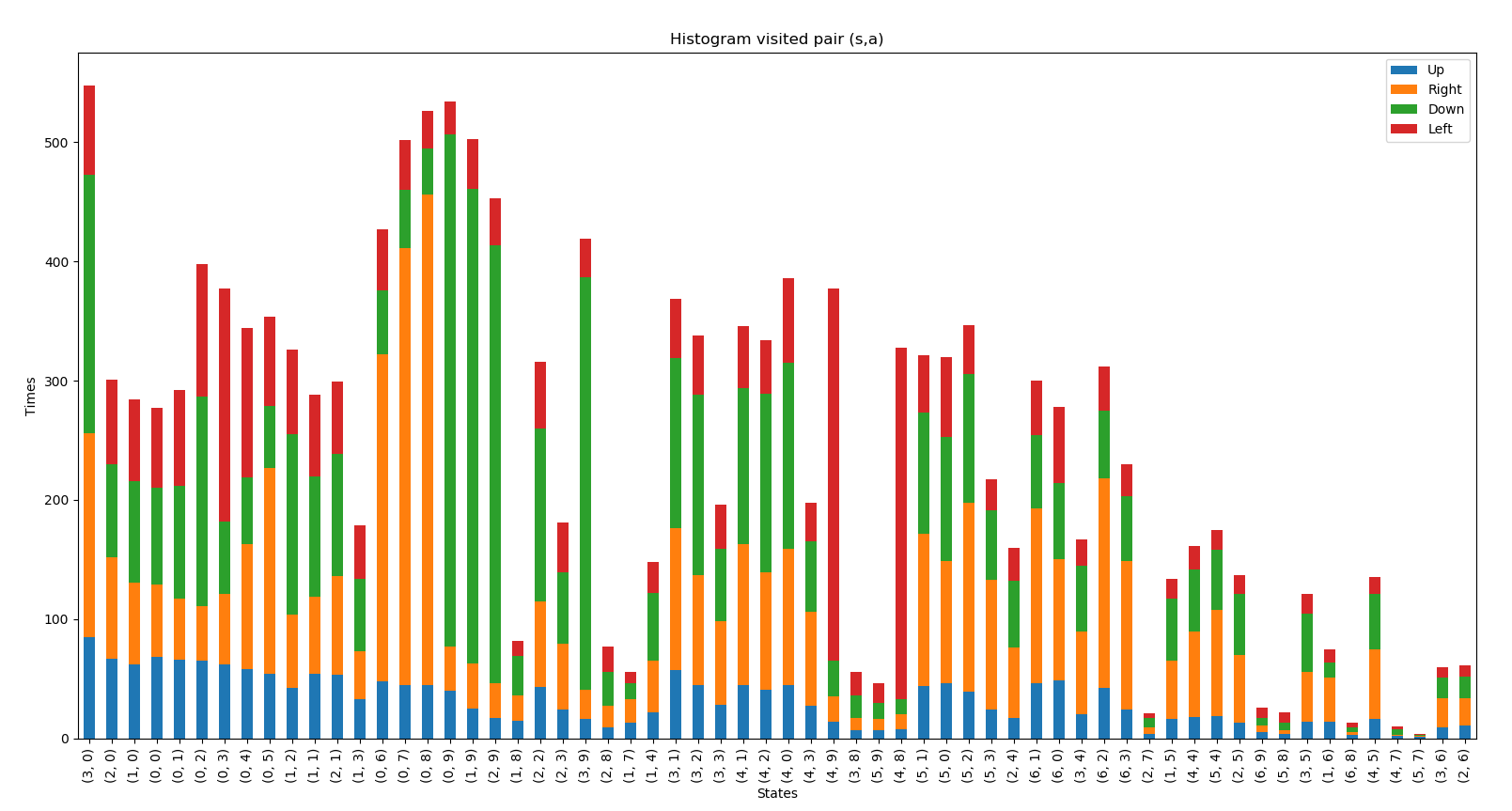
**First:** Sweeping Alpha (Learning Rate)

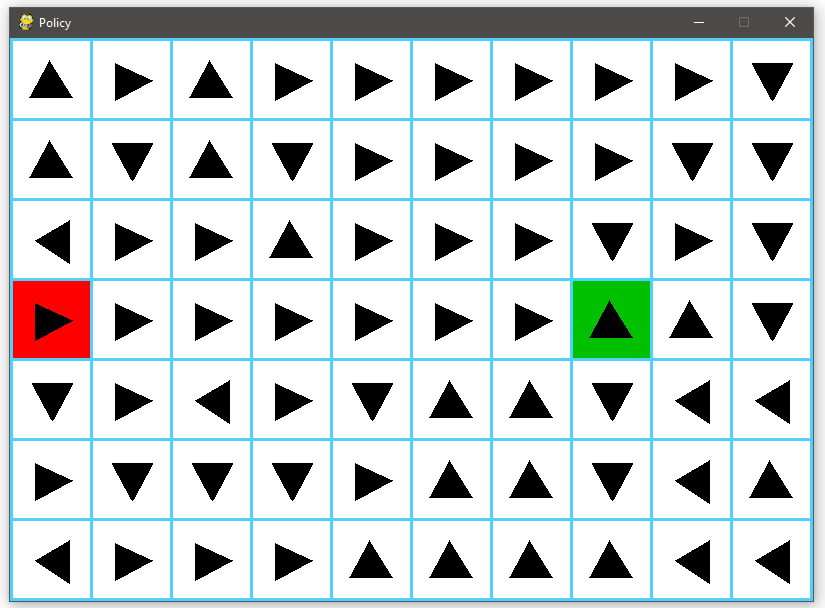
* **Alpha (α):** 0.2
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 300

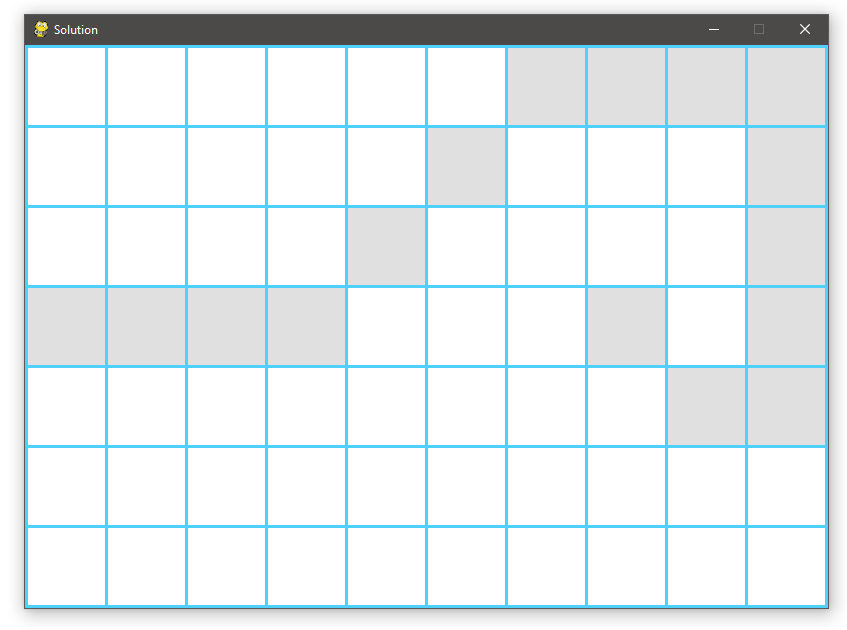
**Minimal number of Episodes:** 151, 157, 154, 133, 179

**Average number of Episodes:** 155





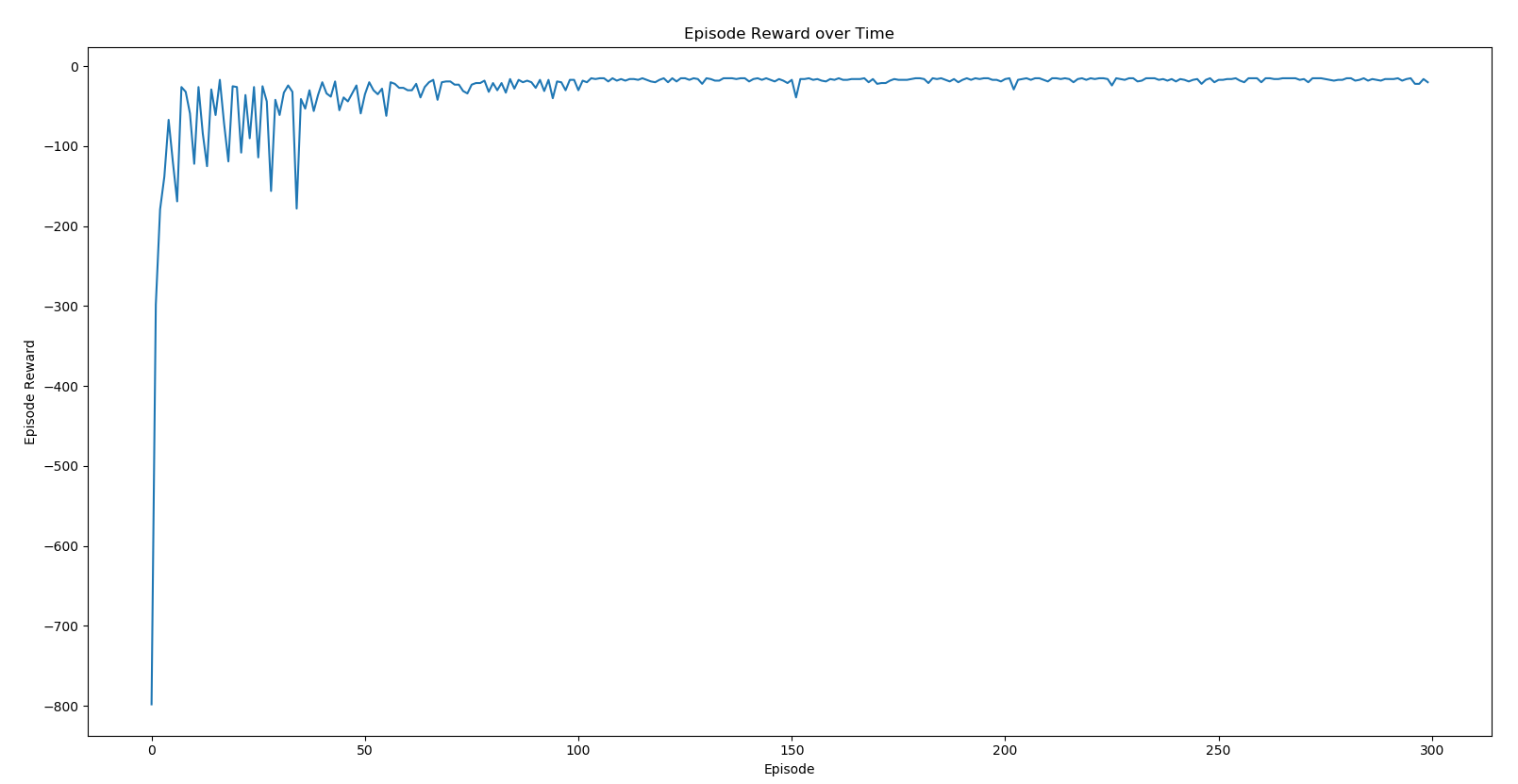


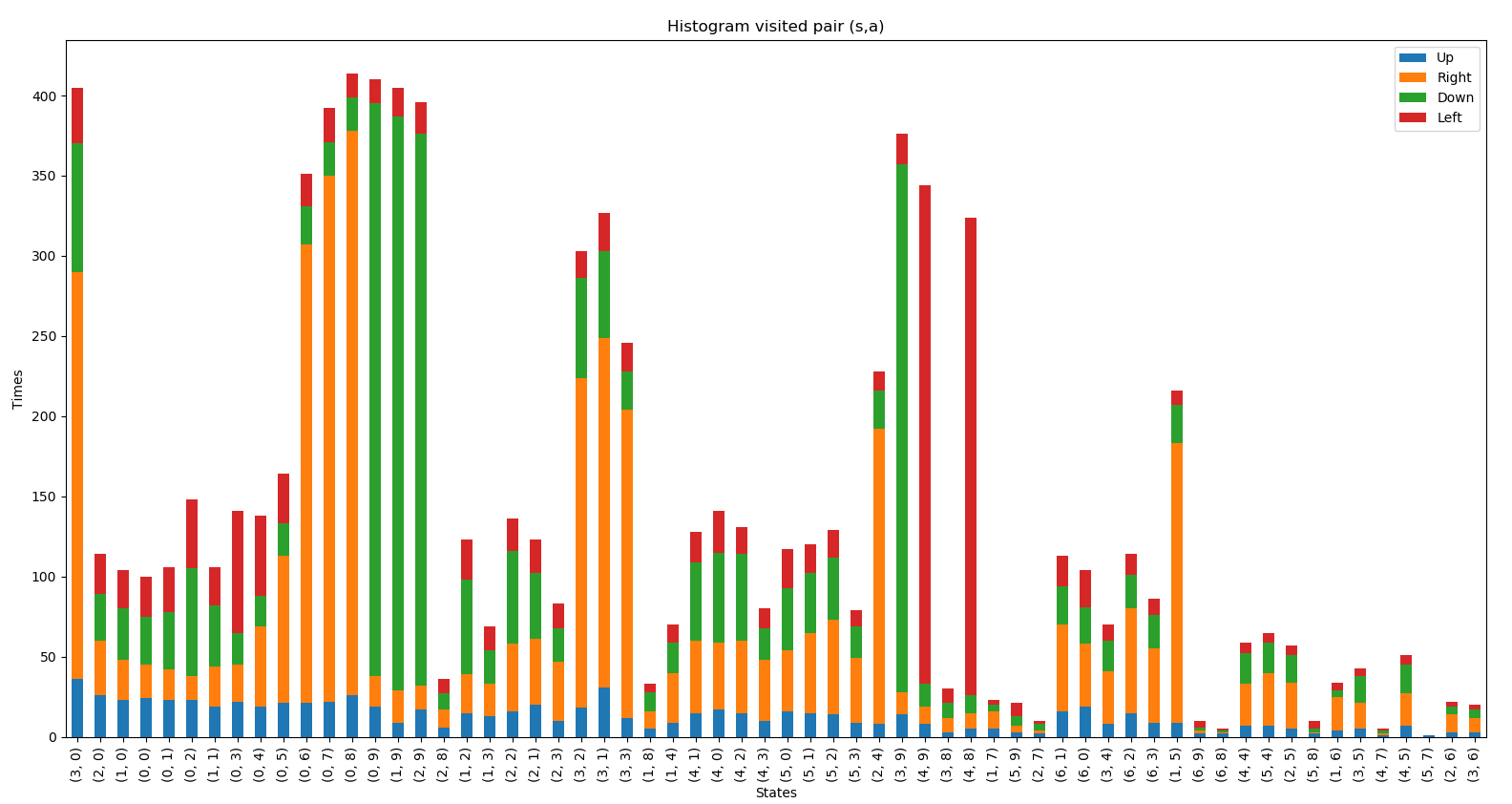


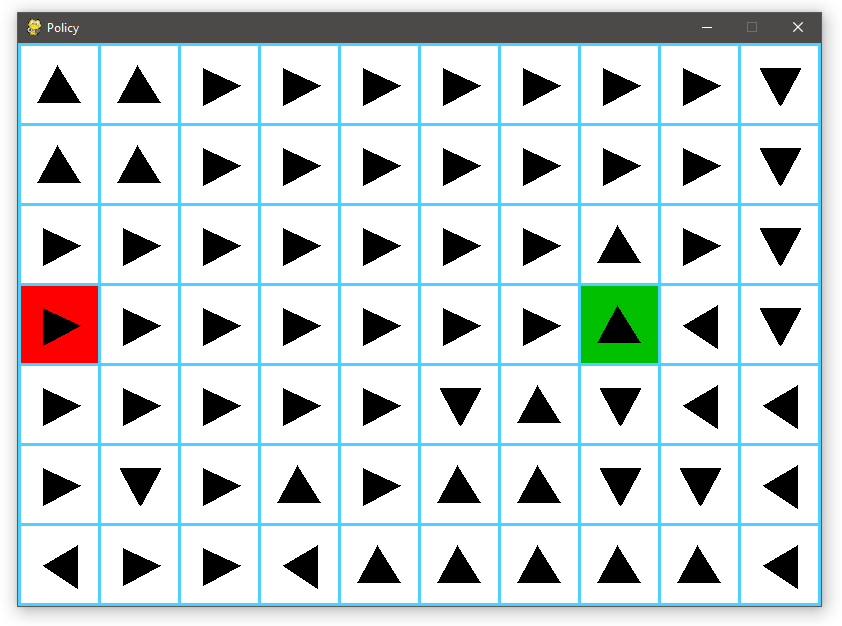
* **Alpha (α):** 0.6
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 300

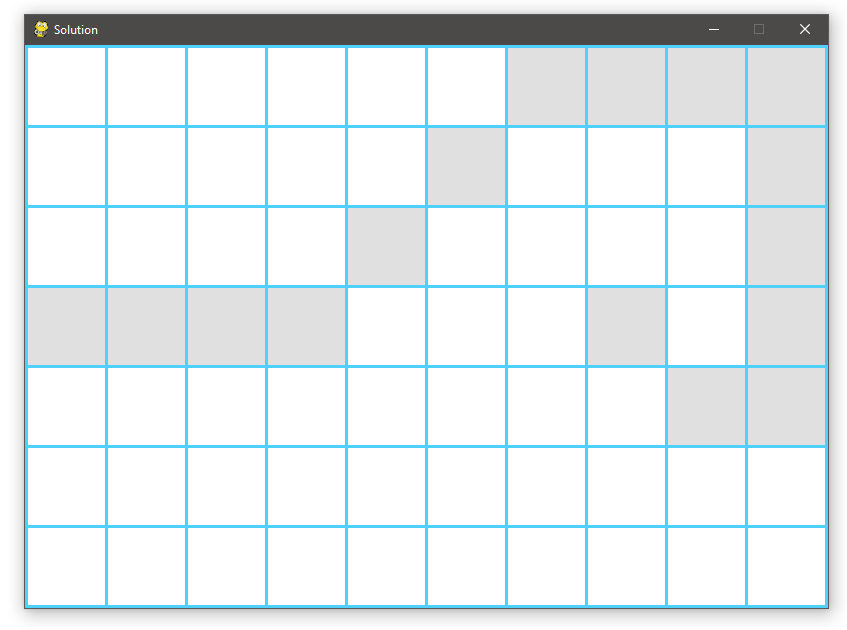
**Minimal number of Episodes:** 59, 42, 58, 52, 46

**Average number of Episodes:** 51





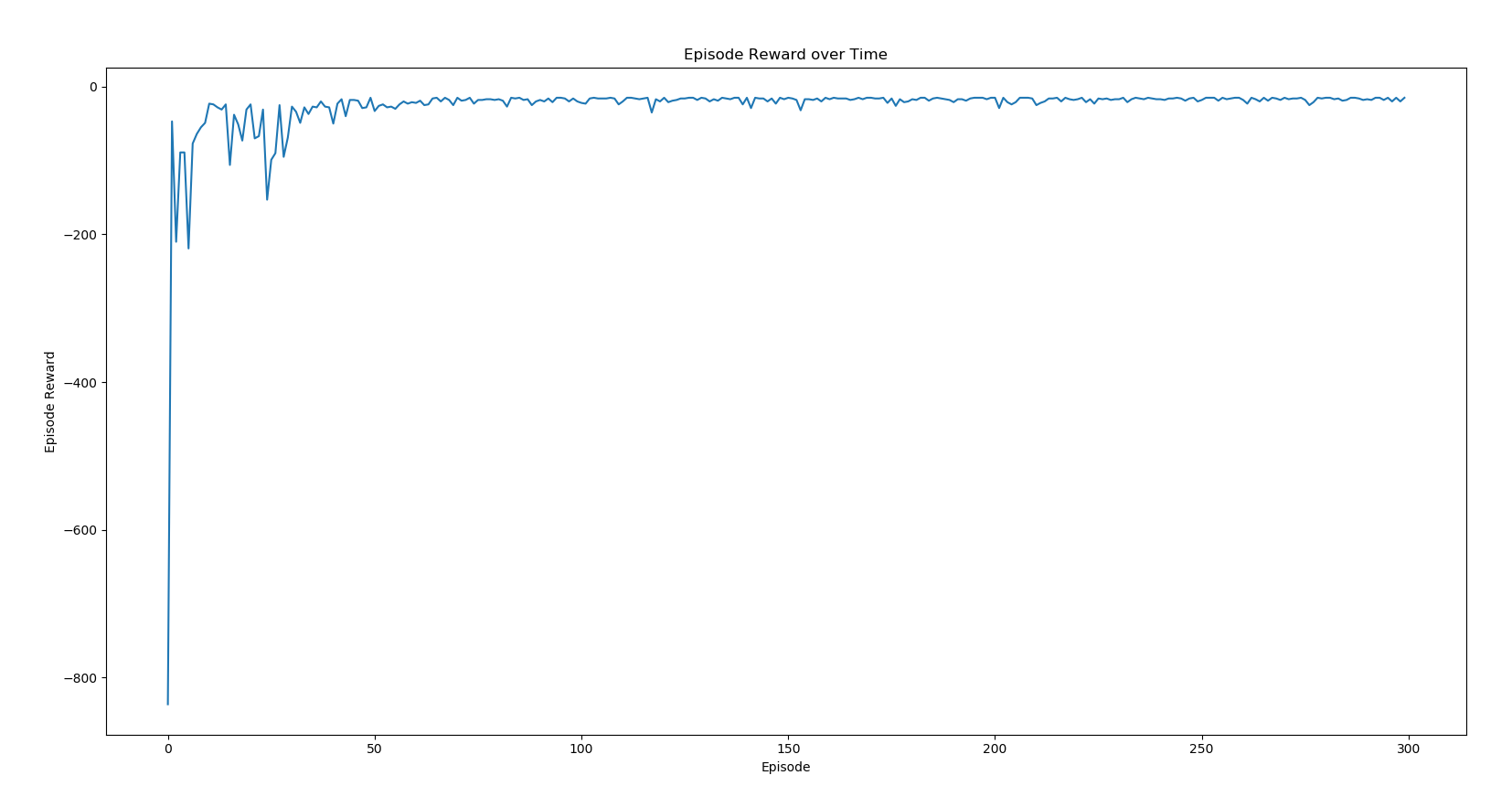


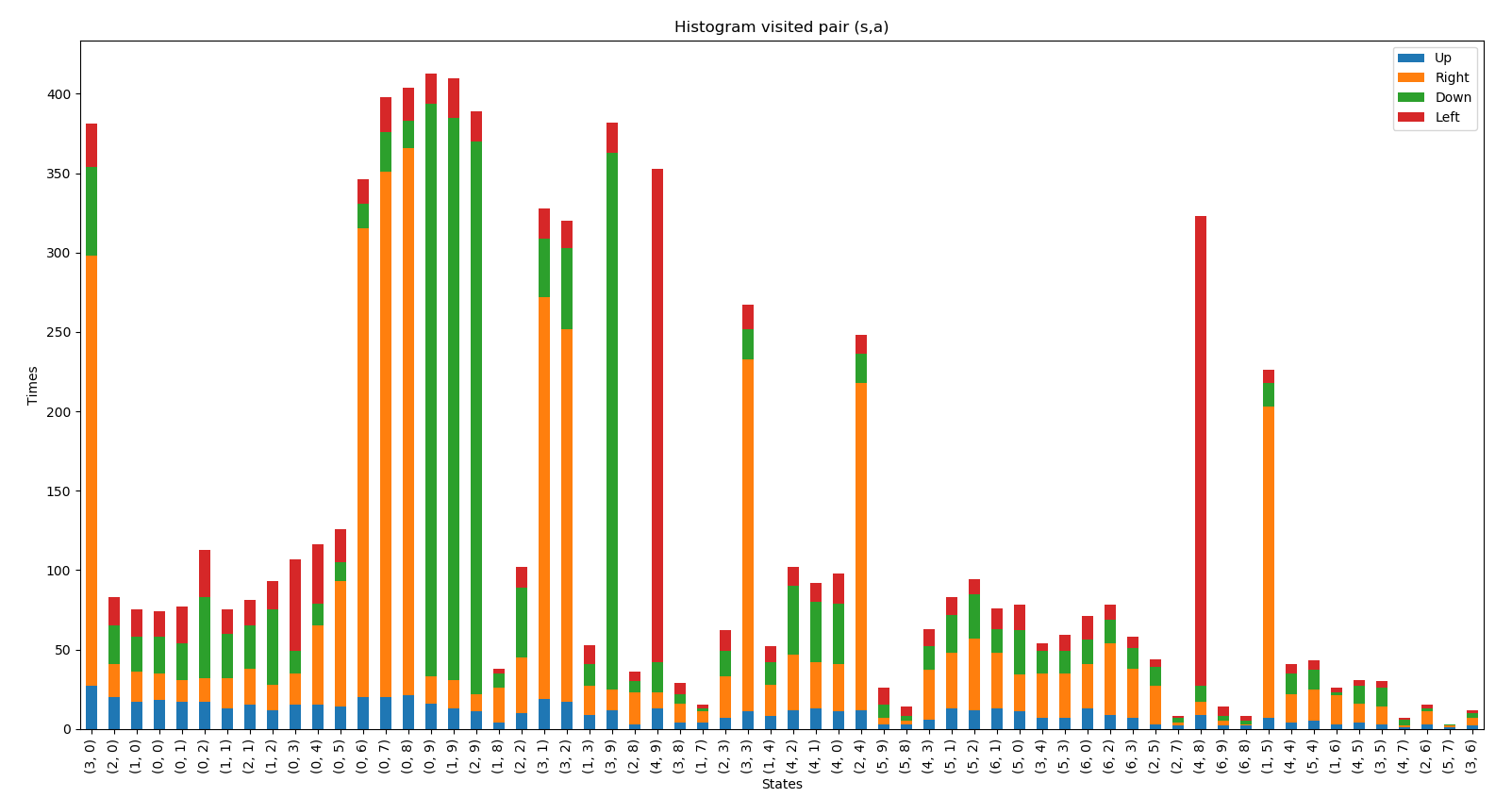


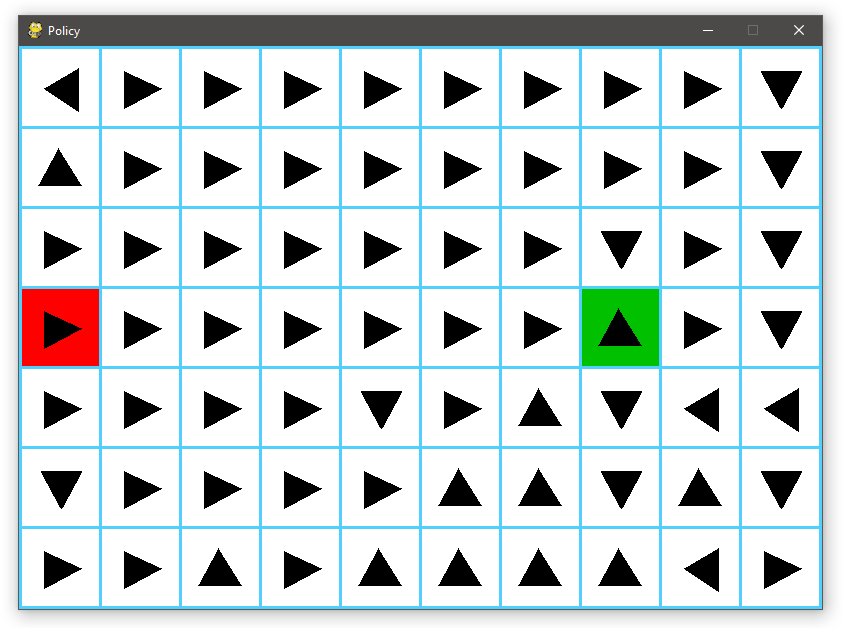
* **Alpha (α):** 0.9
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 300

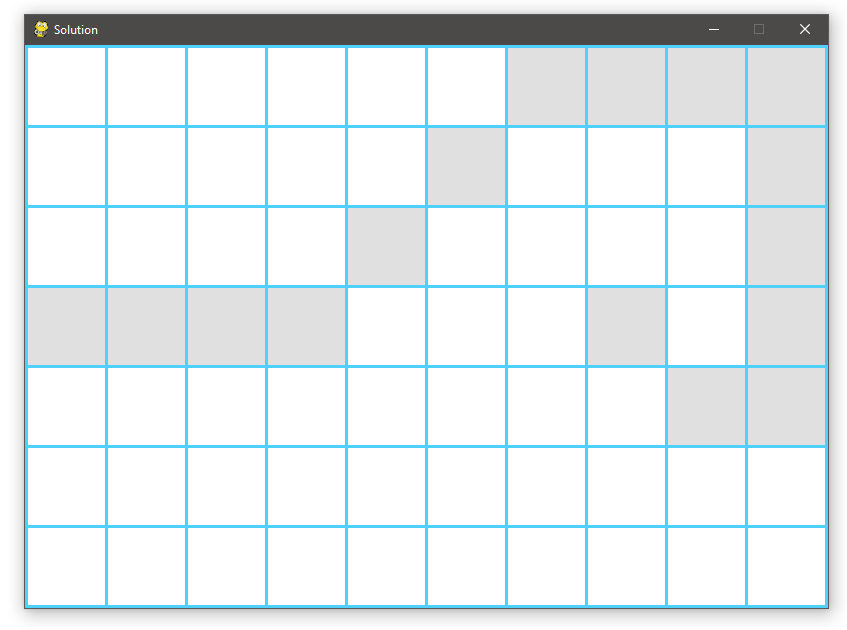
**Minimal number of Episodes:** 50, 58, 37, 62, 55

**Average number of Episodes:** 52







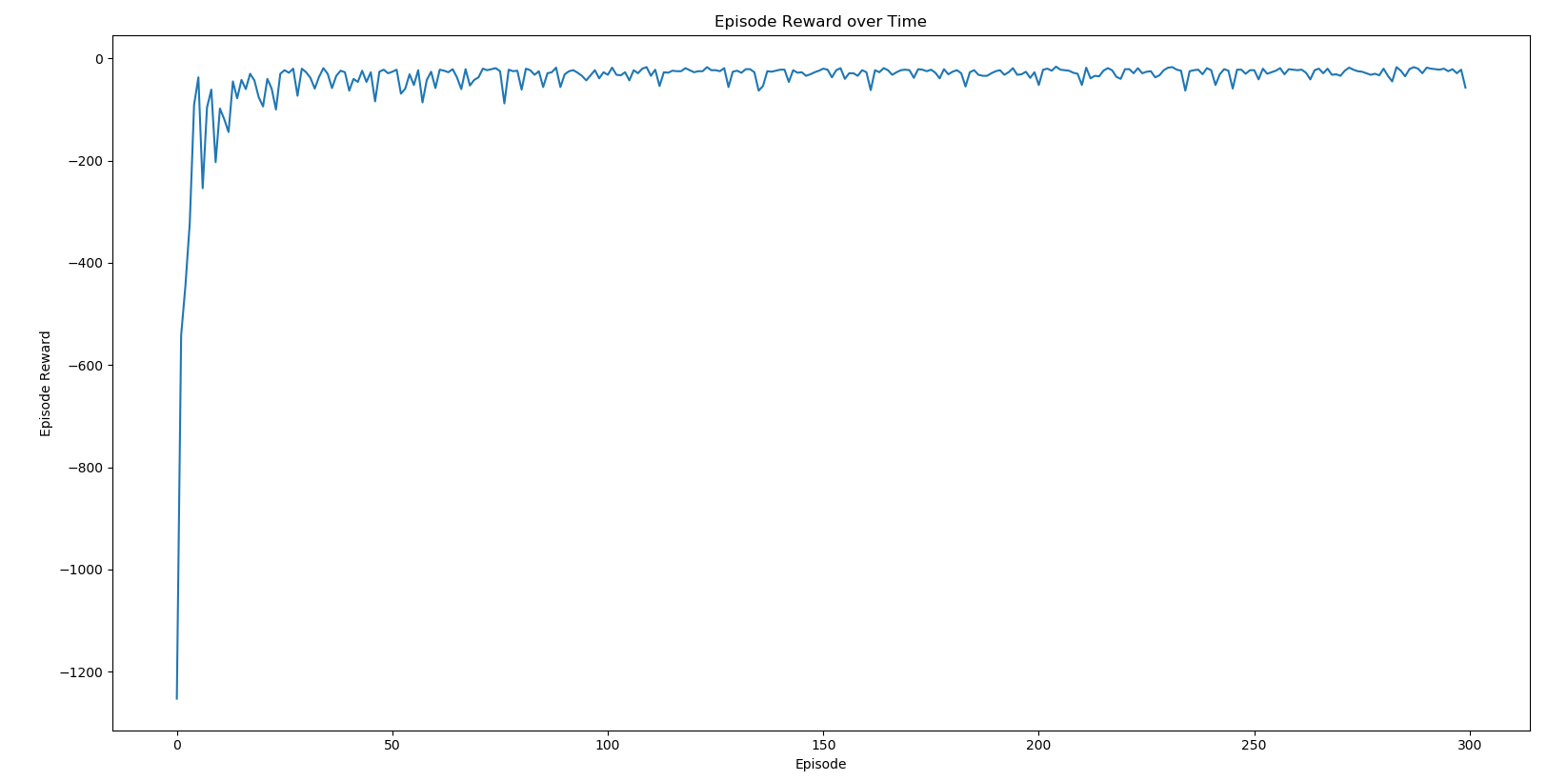


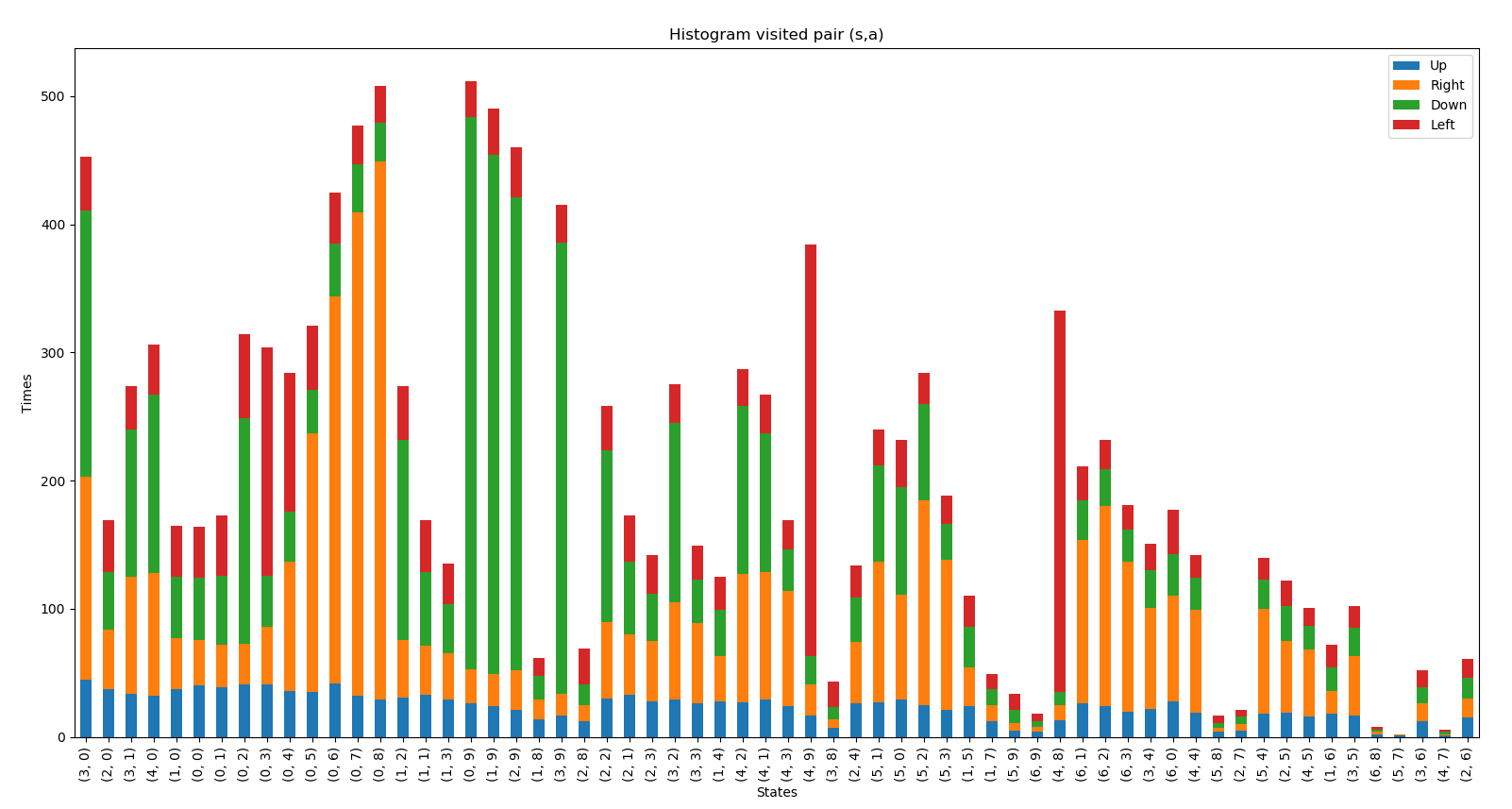
**Second:** Sweeping Gamma (Discounting Rate)

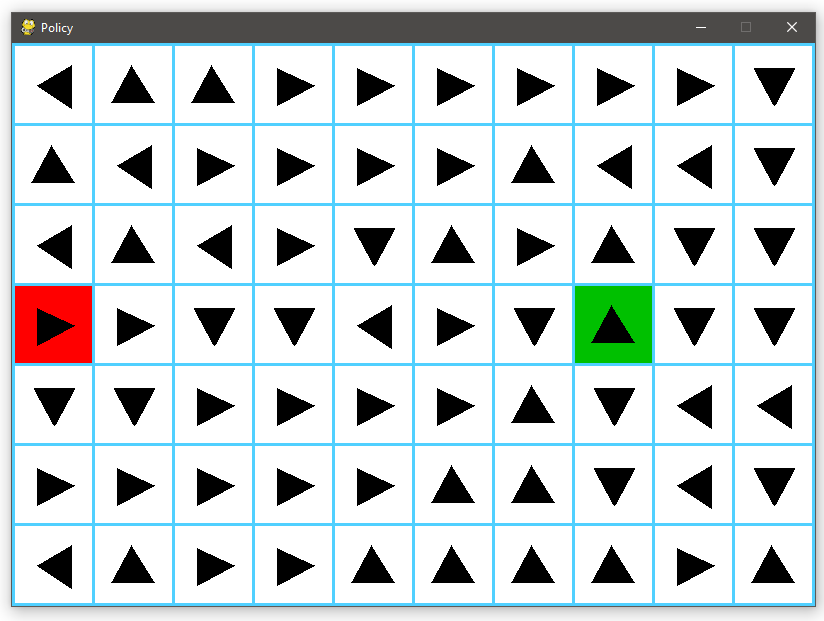
* **Alpha (α):** 0.5
* **Gamma (γ):** 0.2
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 300

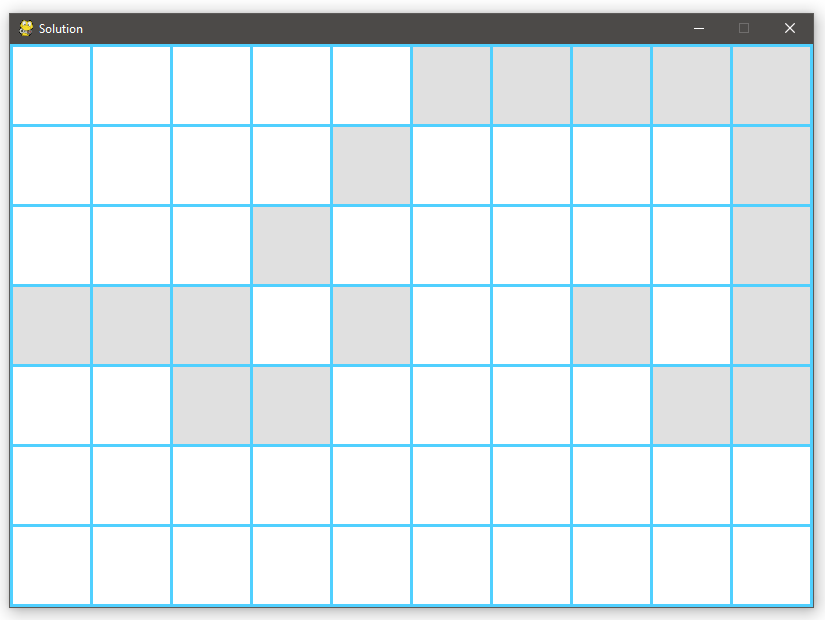
**Minimal number of Episodes:** 78, 55, 39, 51, 61

**Average number of Episodes:** 56





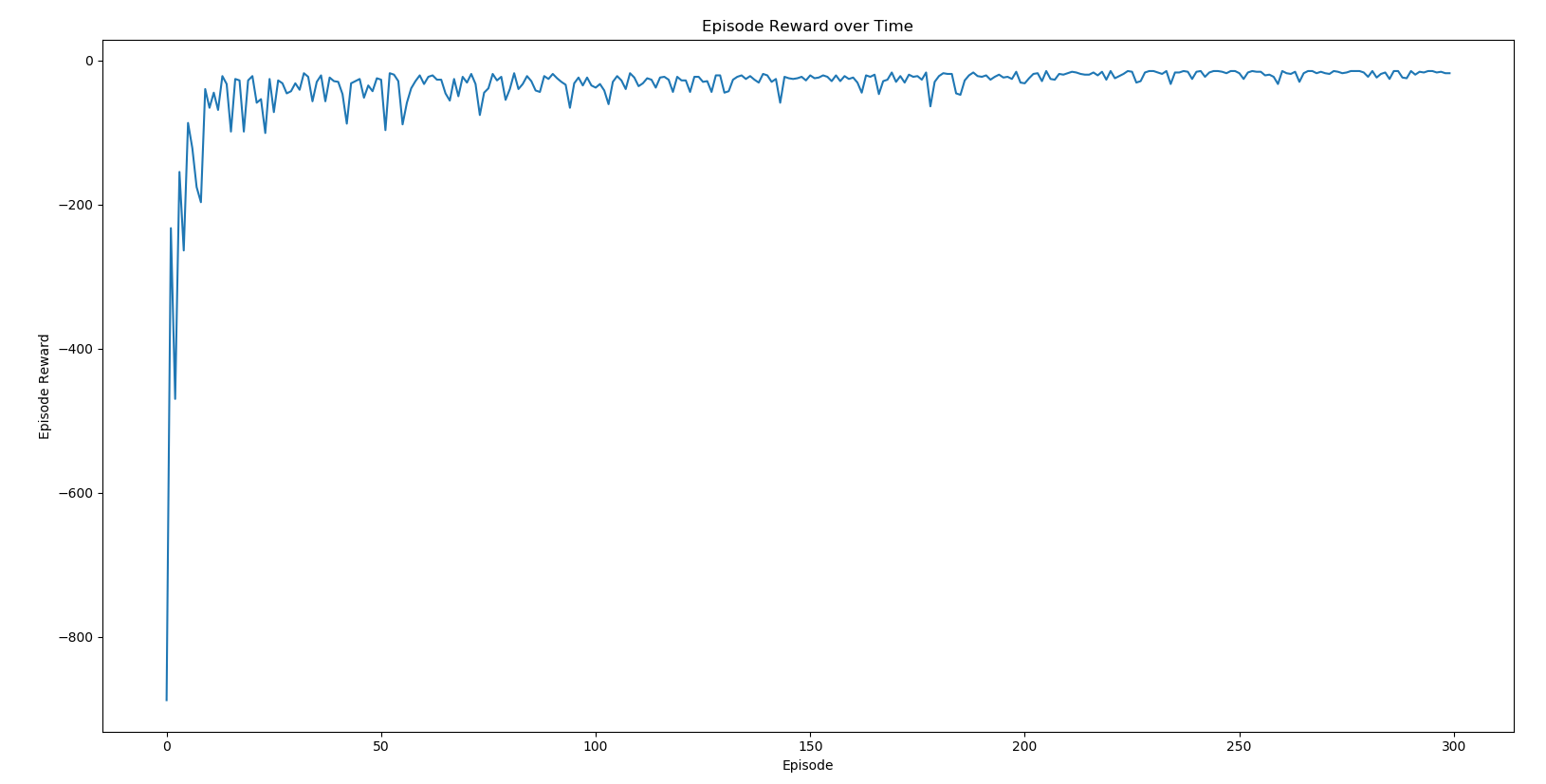


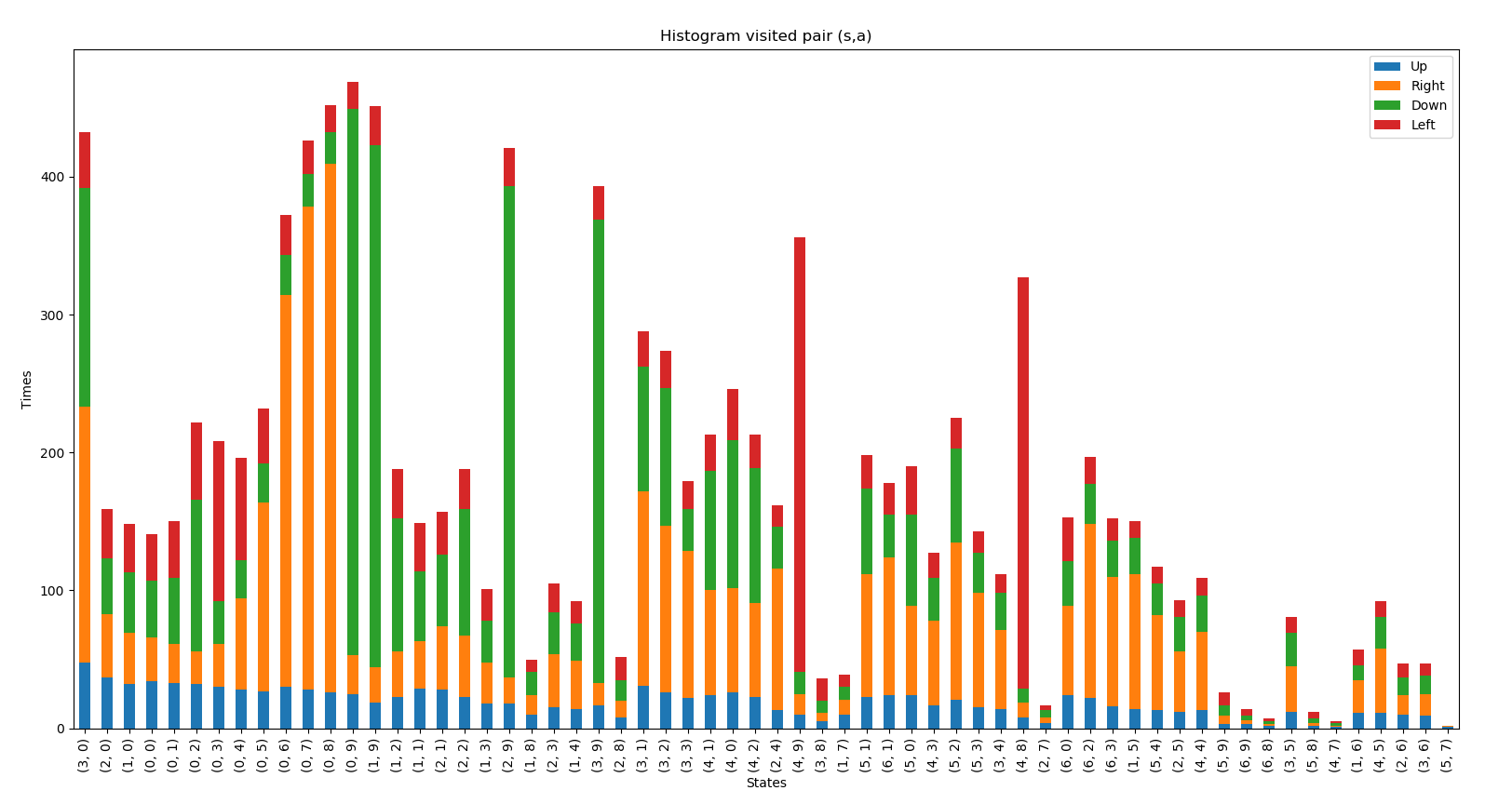


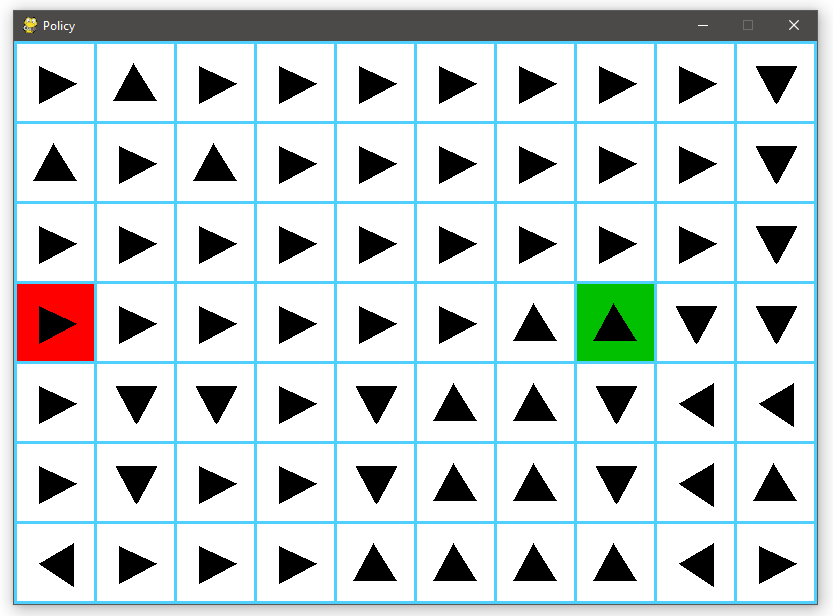
* **Alpha (α):** 0.5
* **Gamma (γ):** 0.5
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 300

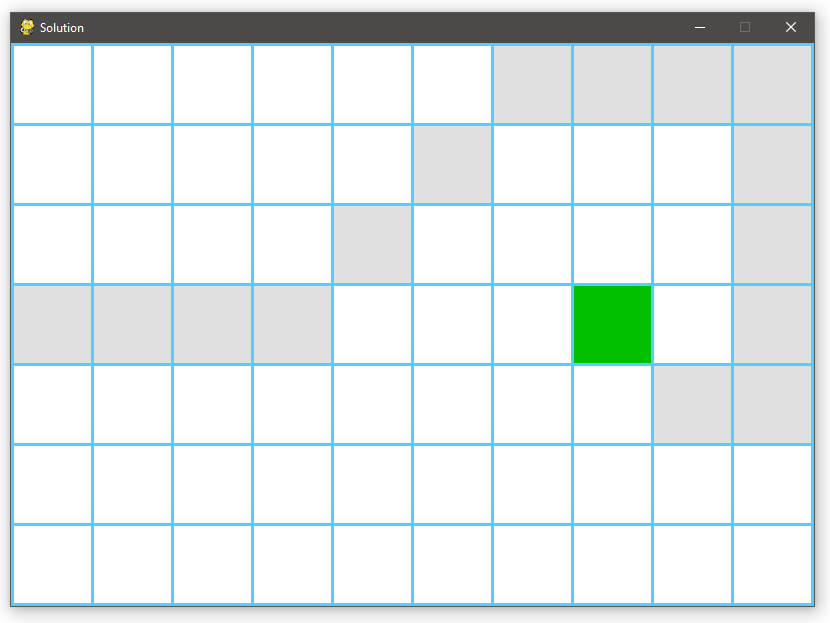
**Minimal number of Episodes:** 57, 43, 37, 45, 46

**Average number of Episodes:** 46





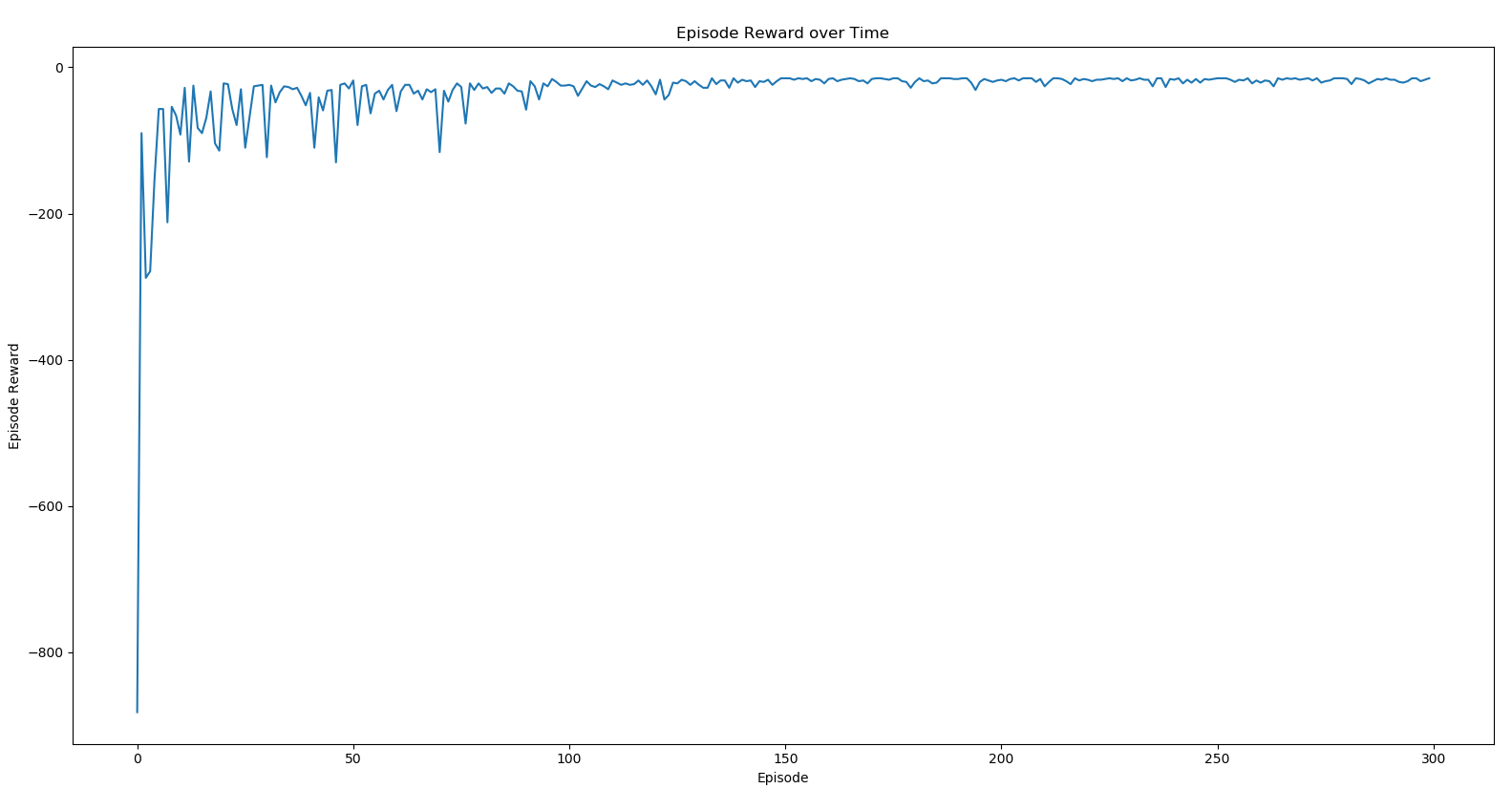


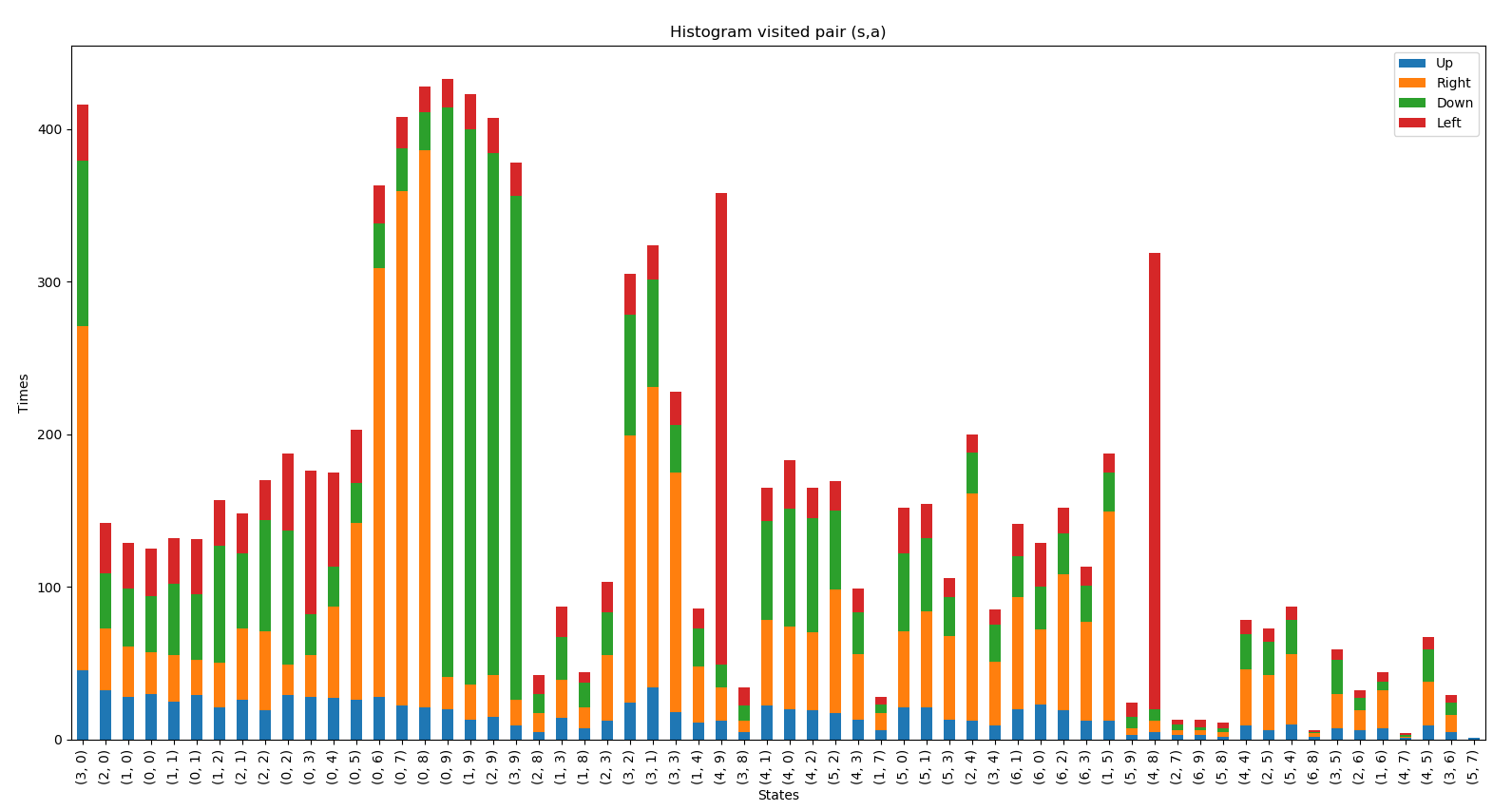


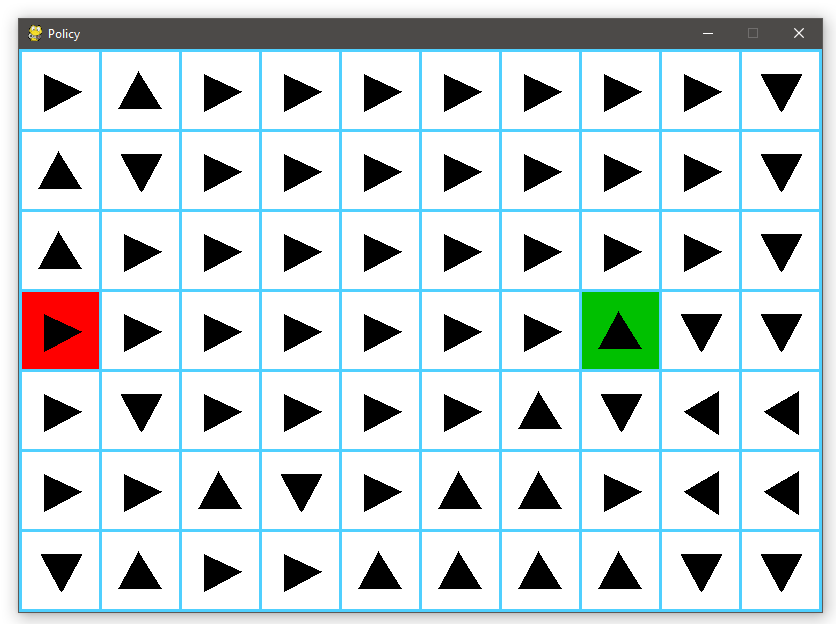
* **Alpha (α):** 0.5
* **Gamma (γ):** 0.8
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 300

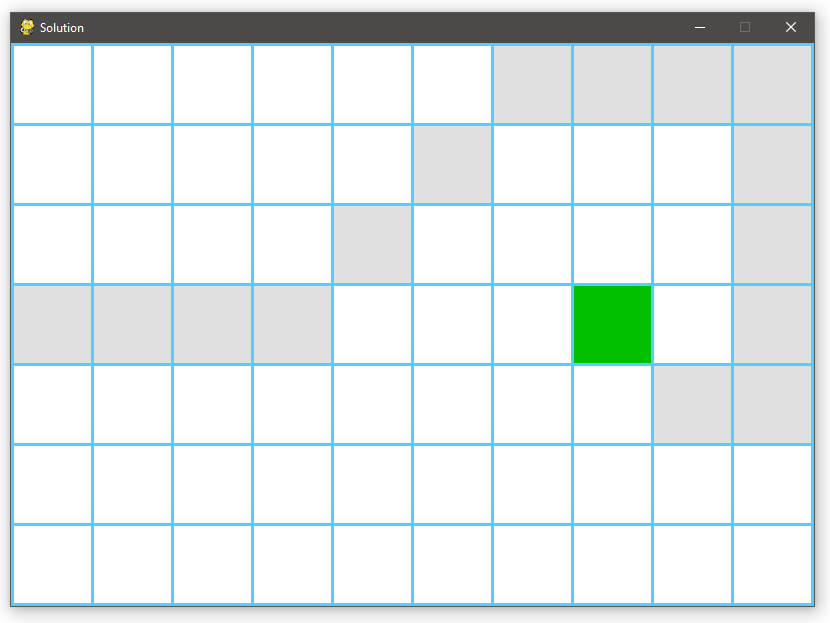
**Minimal number of Episodes:** 72, 70, 102, 79, 70

**Average number of Episodes:** 79







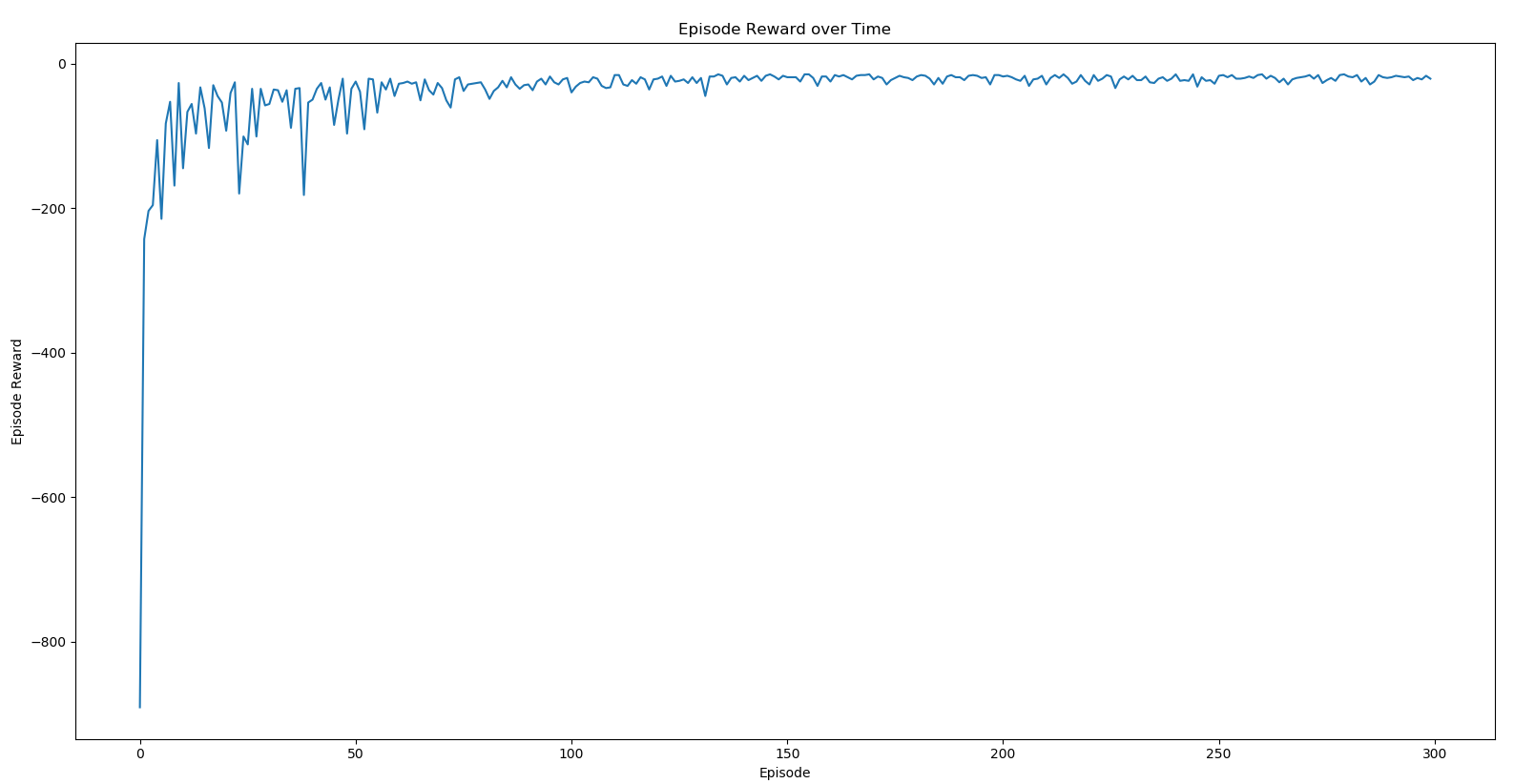


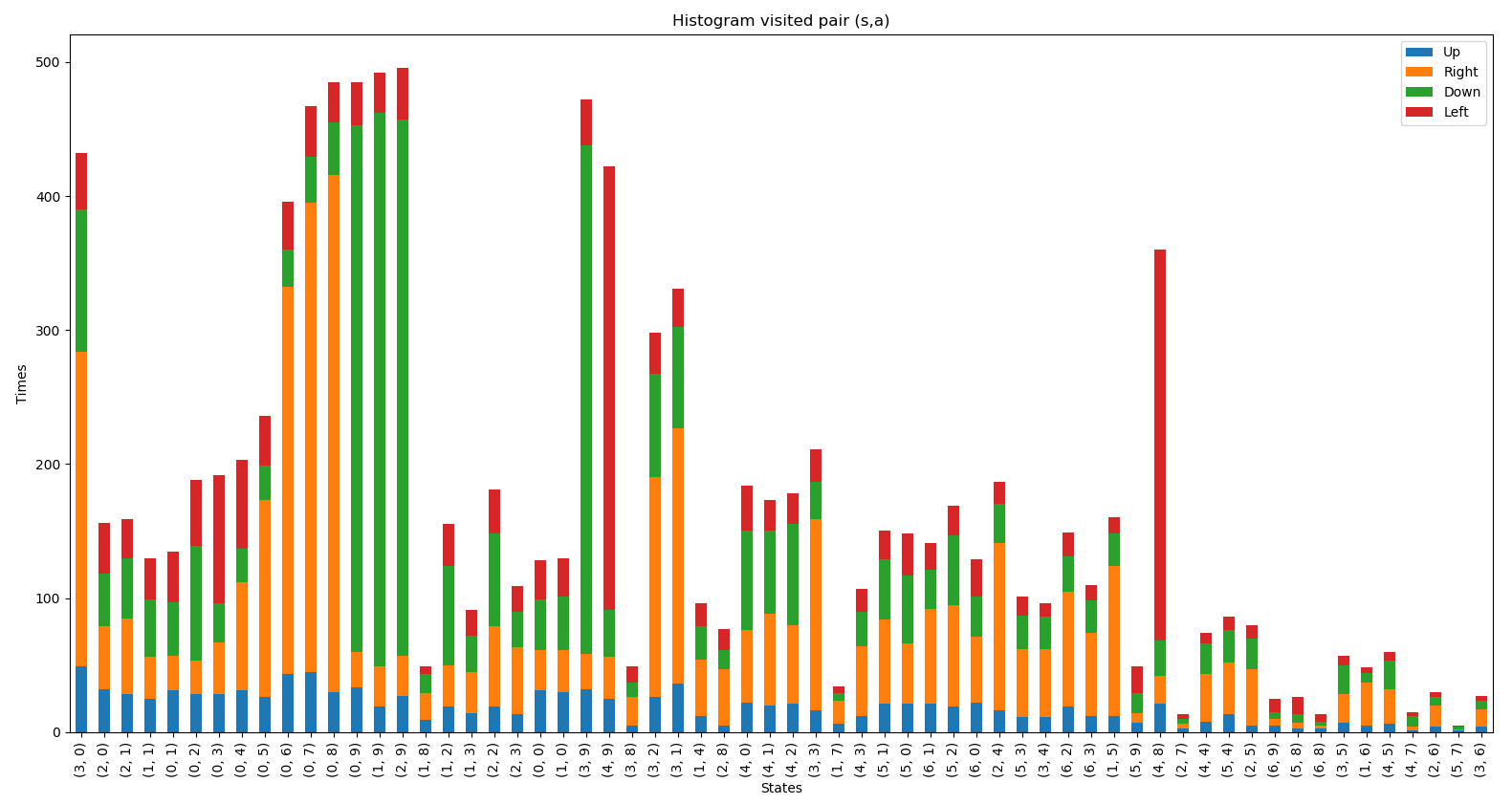
**Third:** Sweeping Epsilon (Exploring Rate)

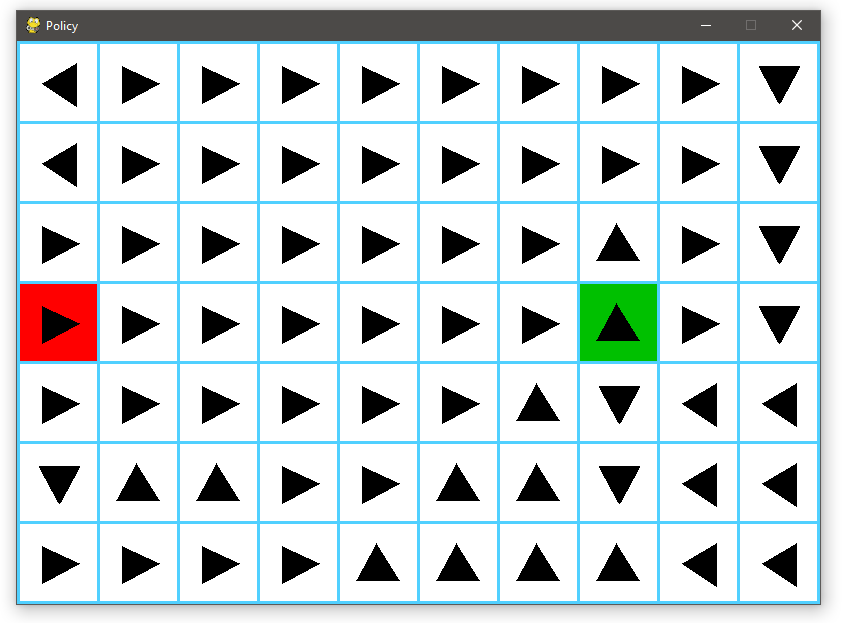
* **Alpha (α):** 0.5
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.2
* **Number of Episodes:** 300

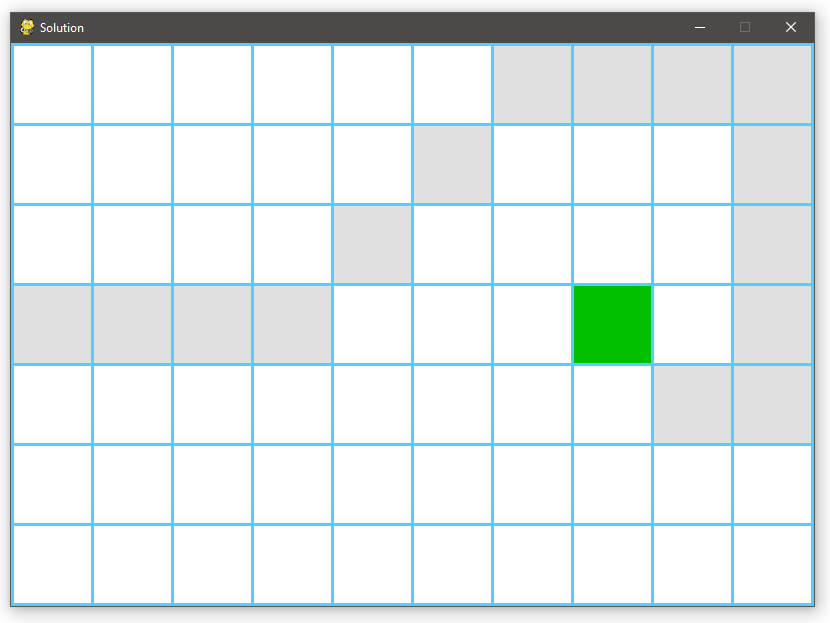
**Minimal number of Episodes:** 48, 76, 43, 66, 72

**Average number of Episodes:** 61





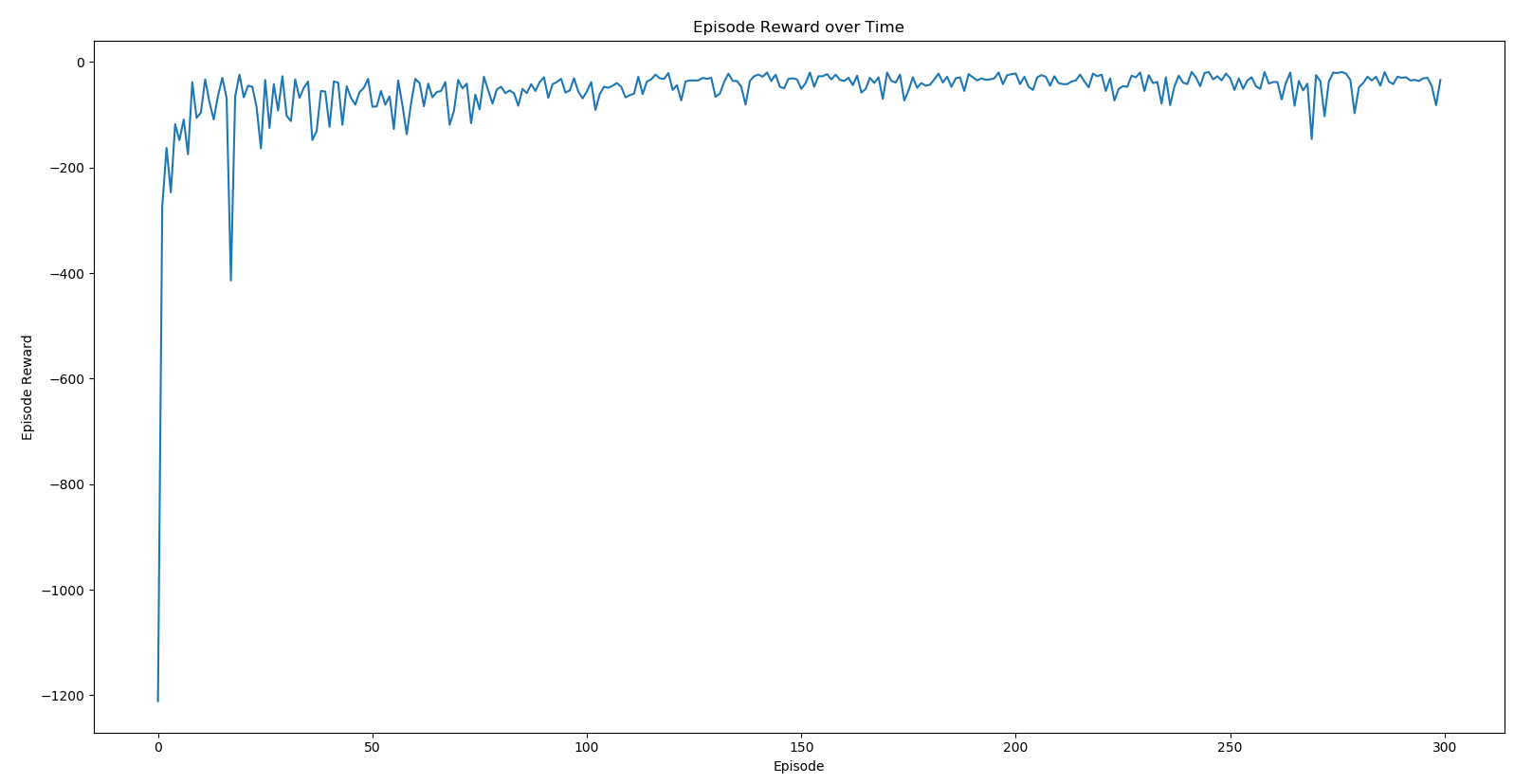


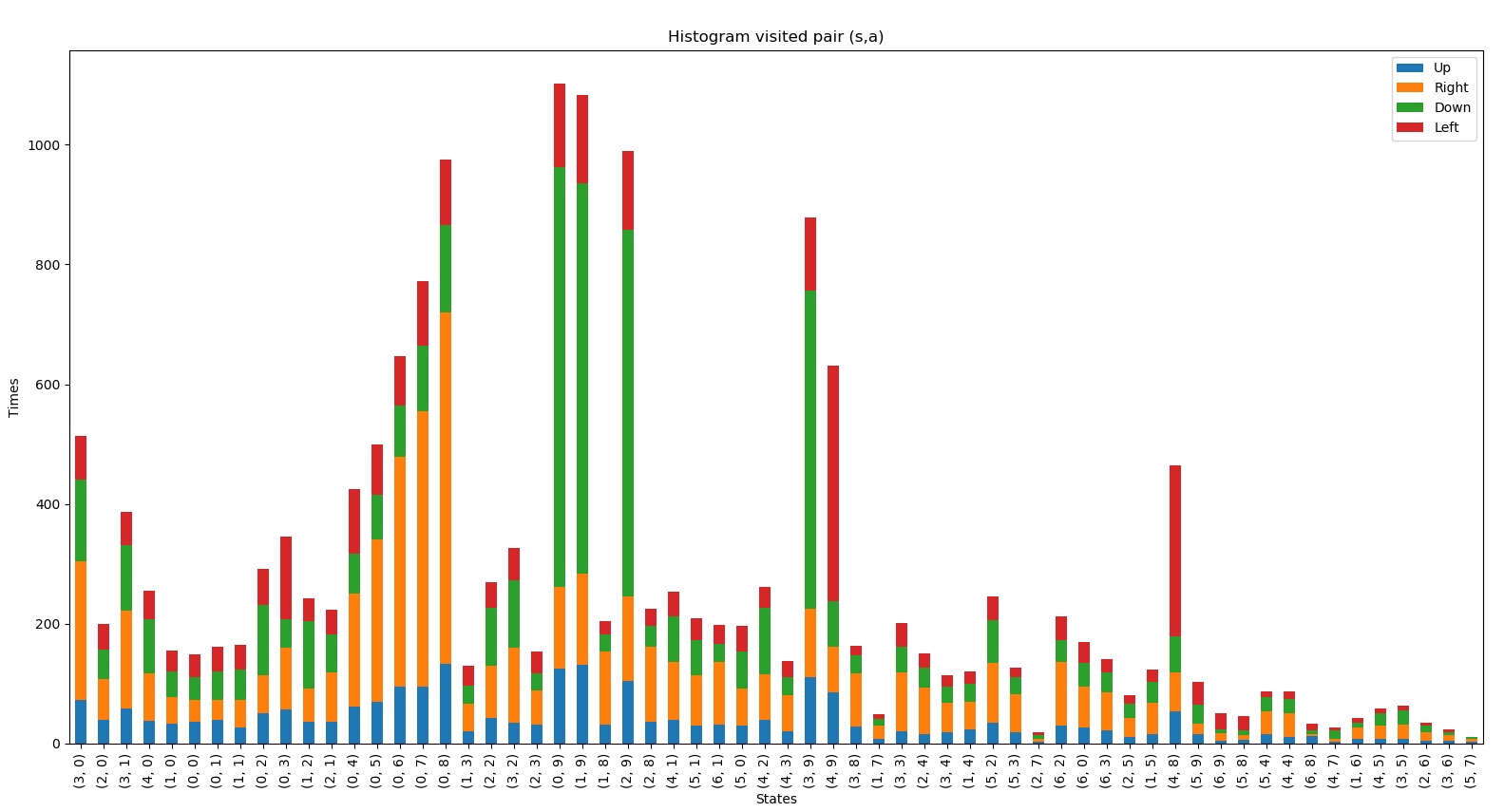


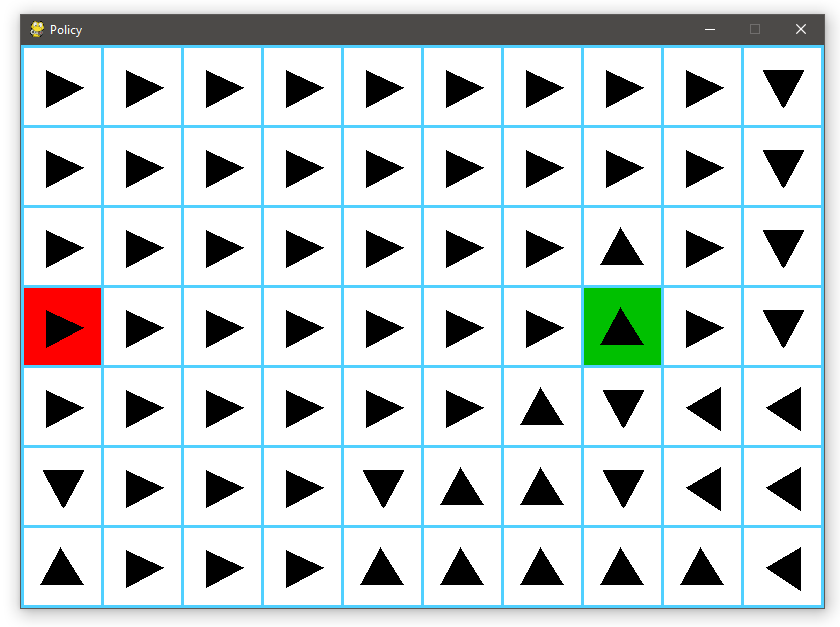
* **Alpha (α):** 0.5
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.5
* **Number of Episodes:** 300

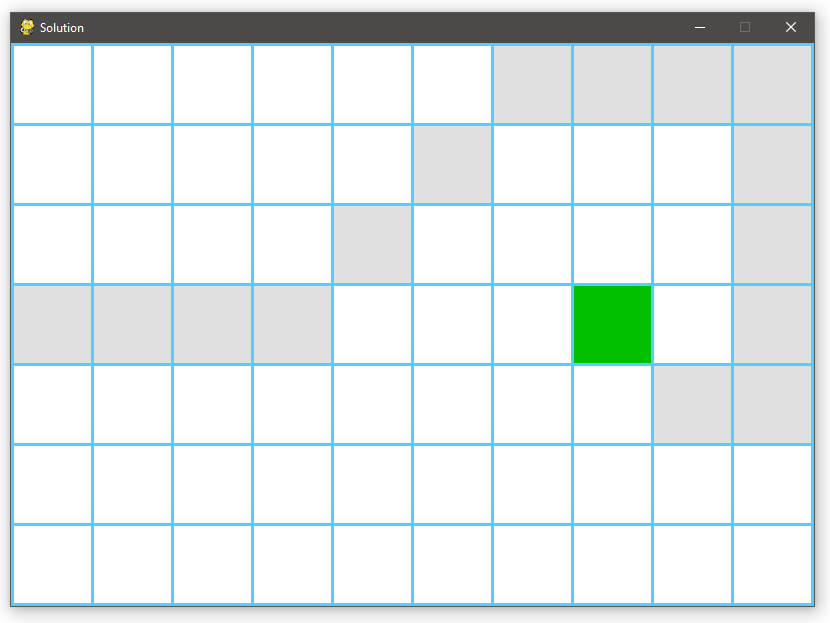
**Minimal number of Episodes:** 37, 31, 30, 32, 36

**Average number of Episodes:** 33





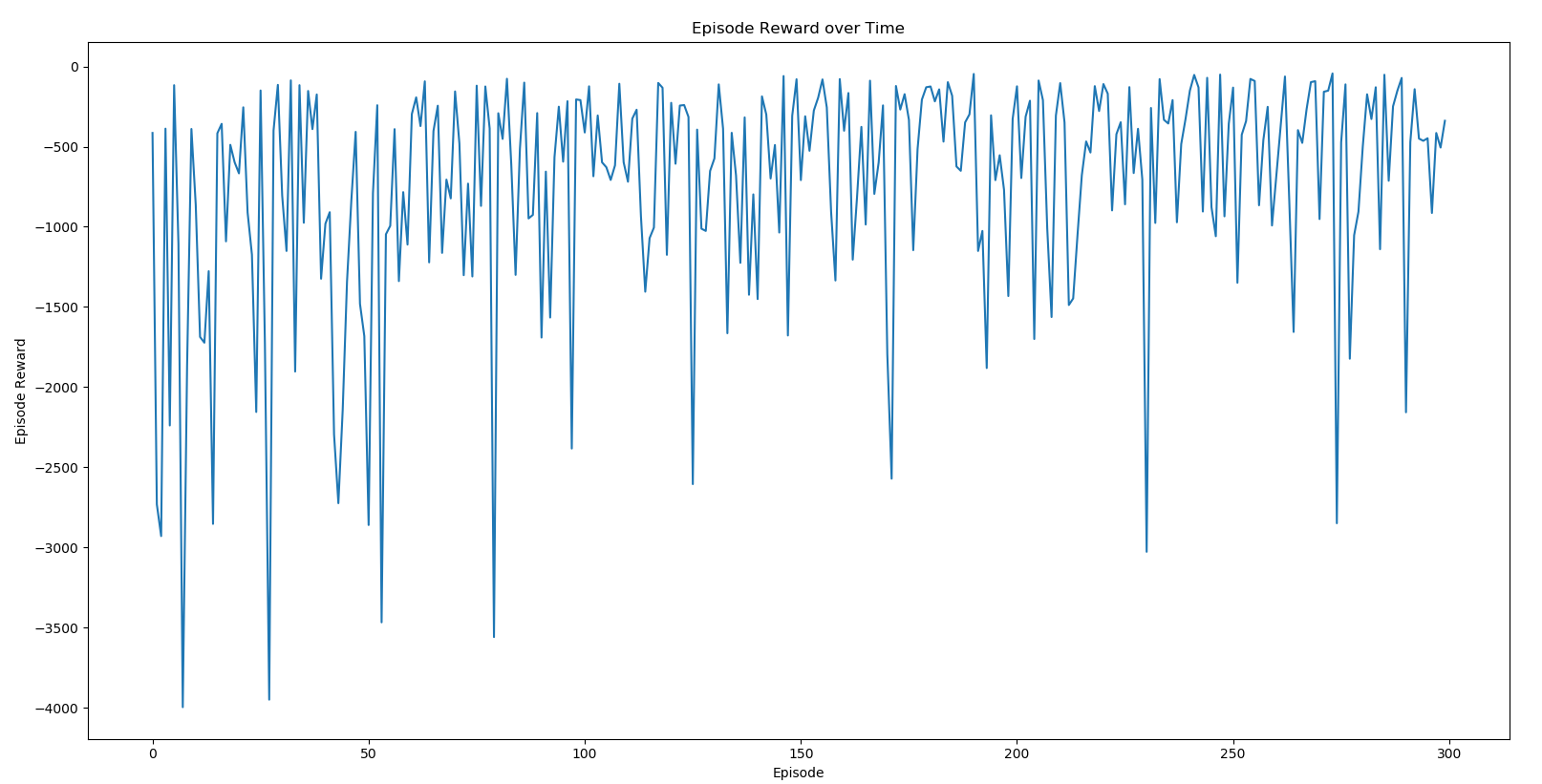


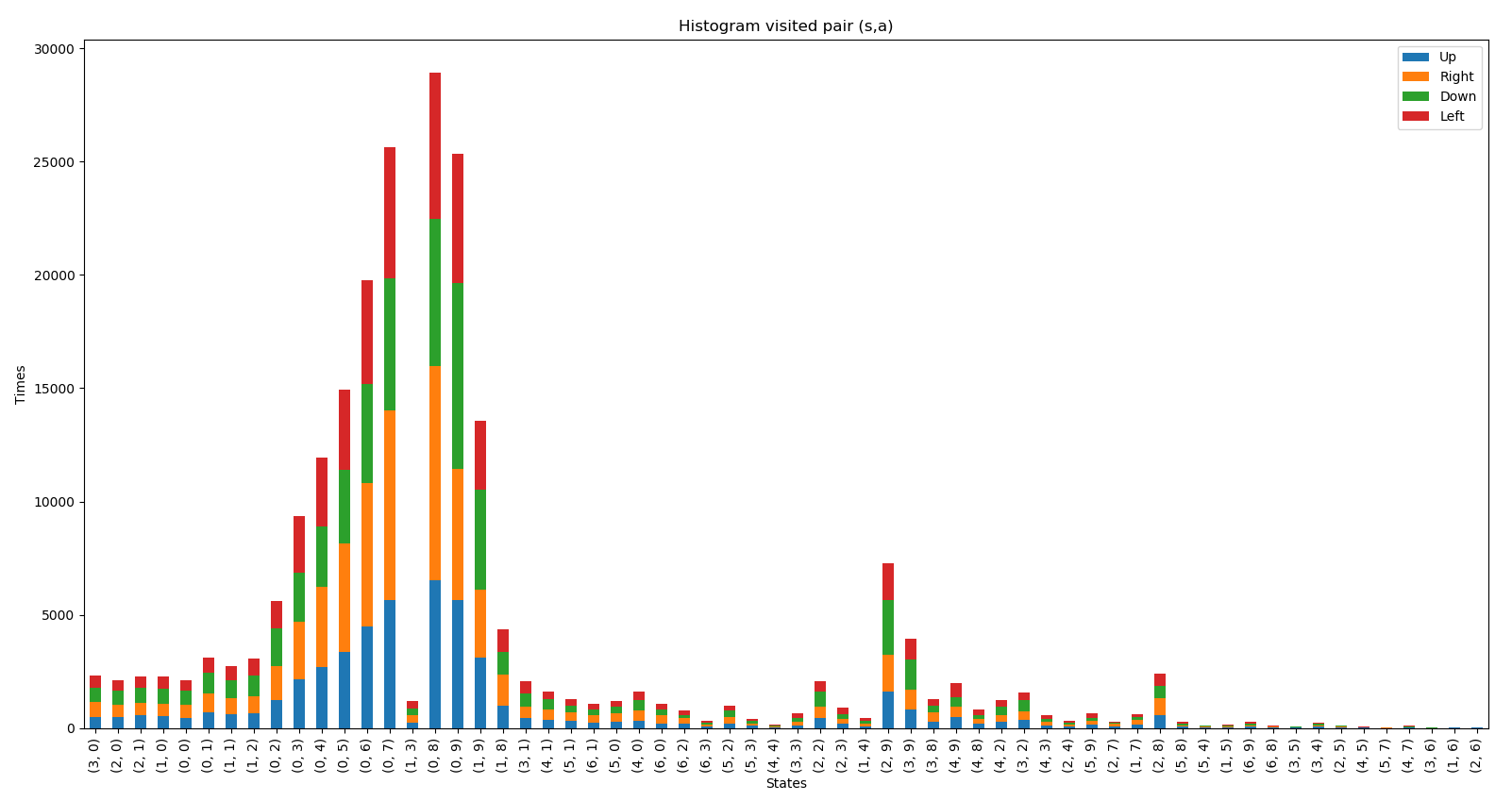


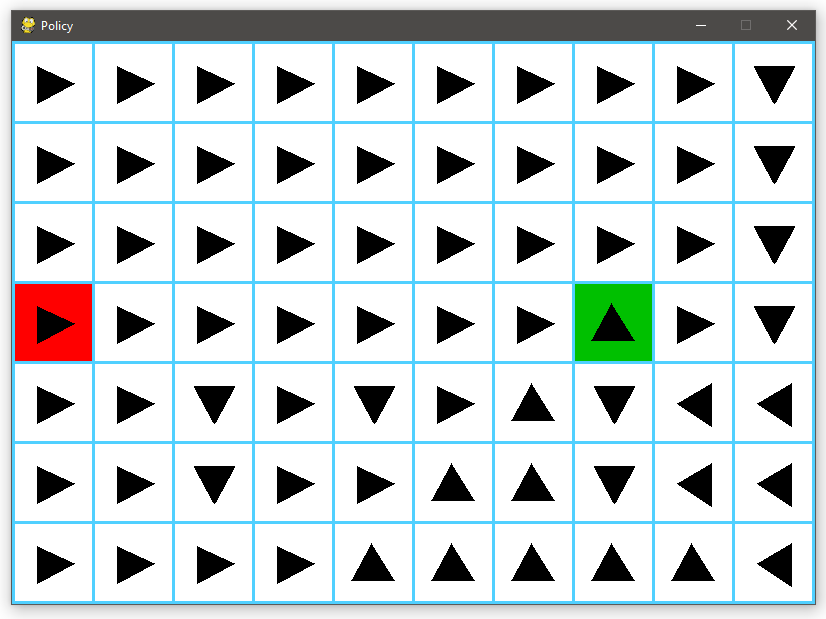
* **Alpha (α):** 0.5
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.9
* **Number of Episodes:** 300

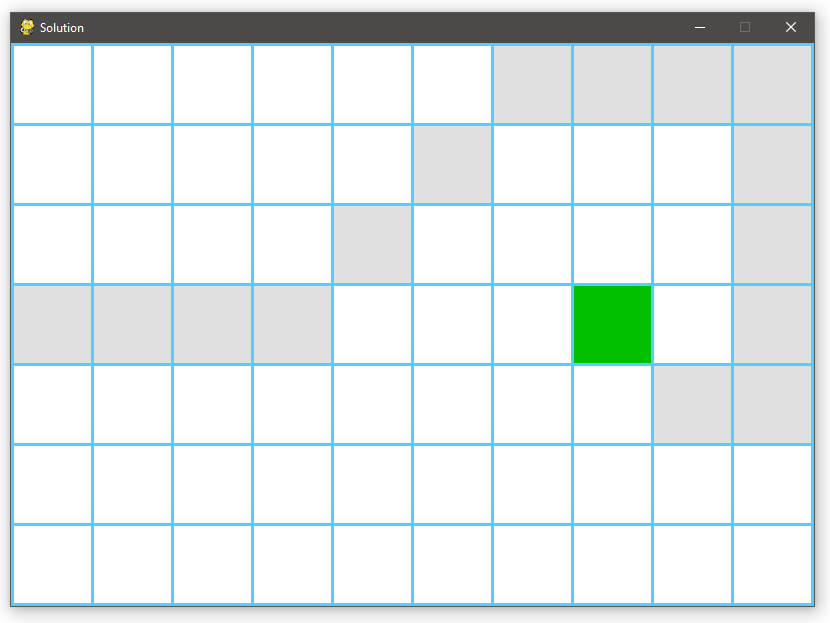
**Minimal number of Episodes:** 129, 416, 276, 432, 100

**Average number of Episodes:** 271



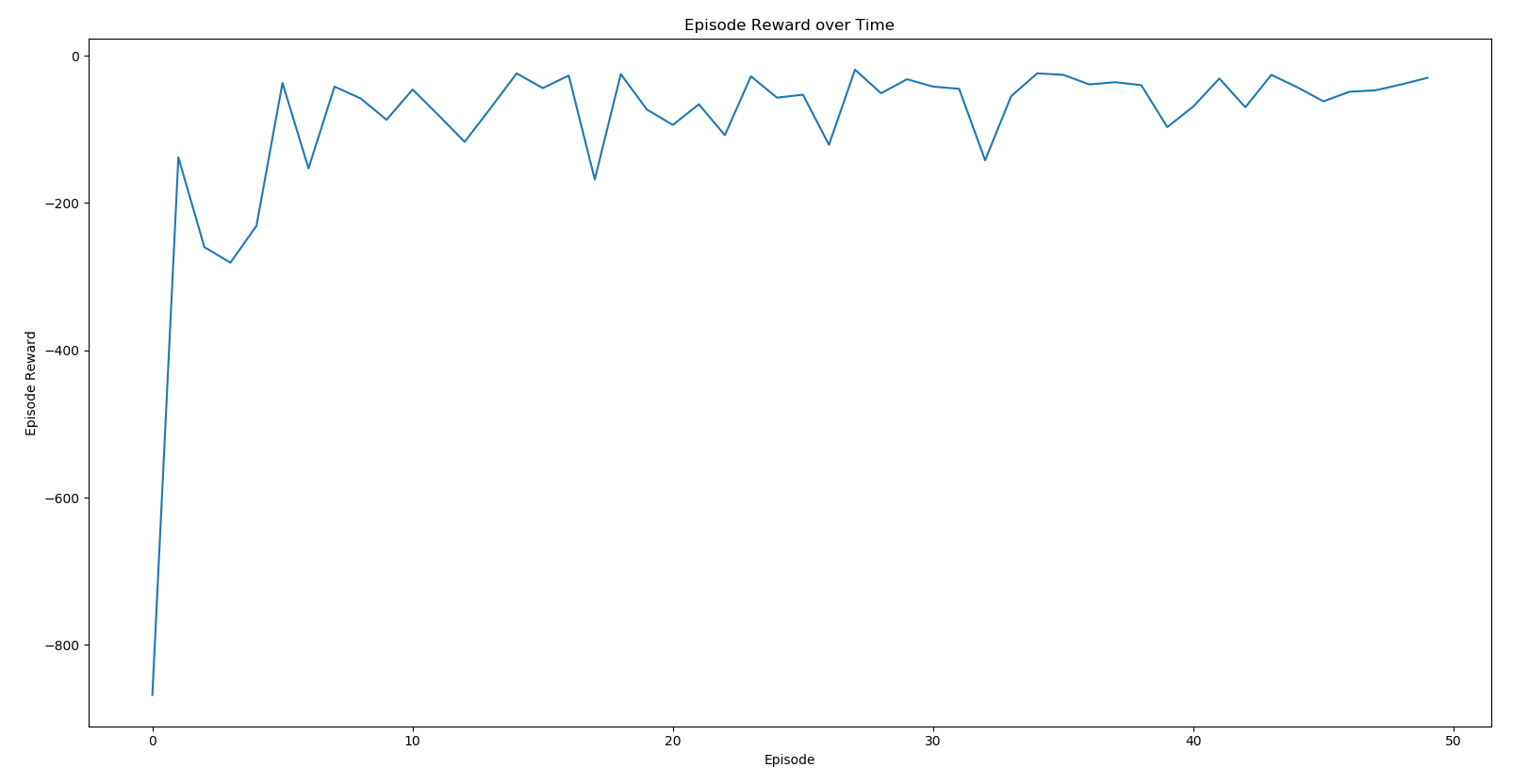


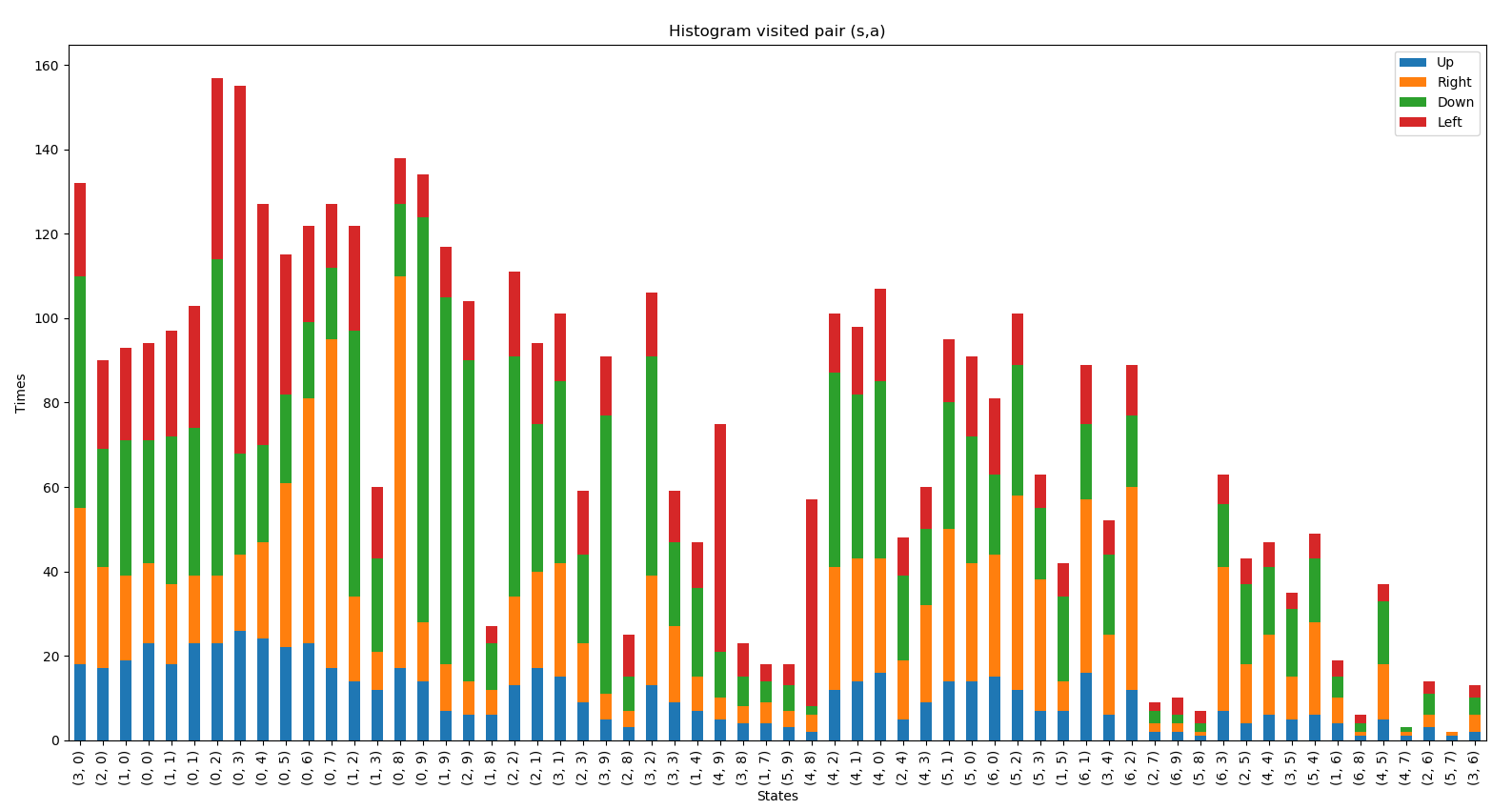


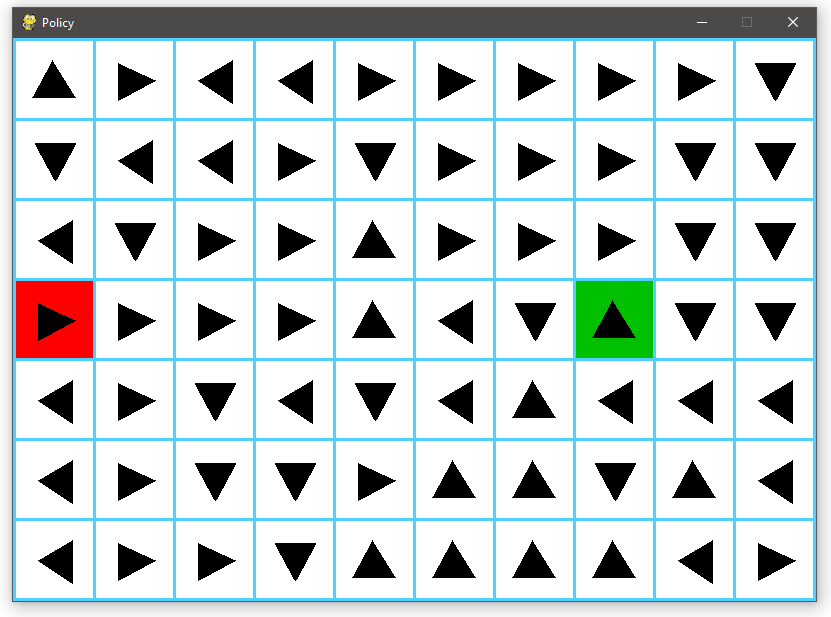
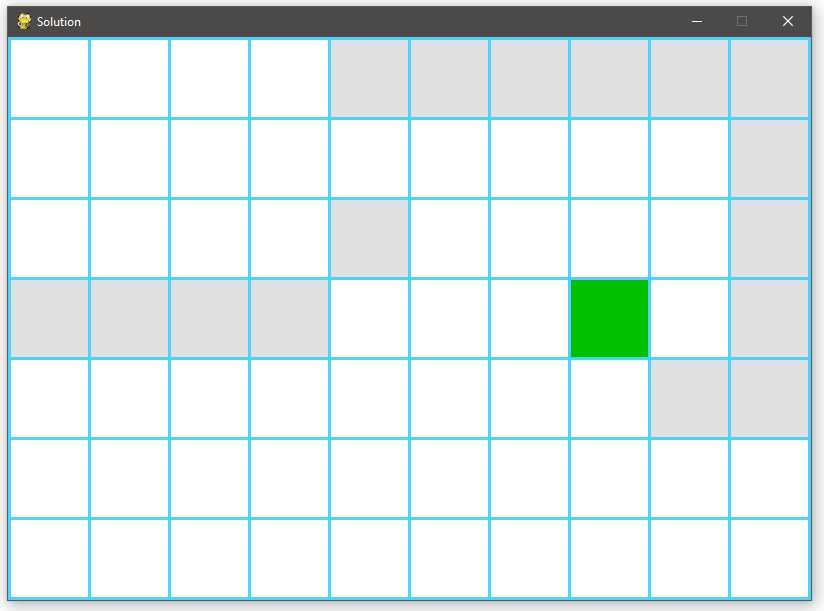


**Fourth:** Sweeping Episodes

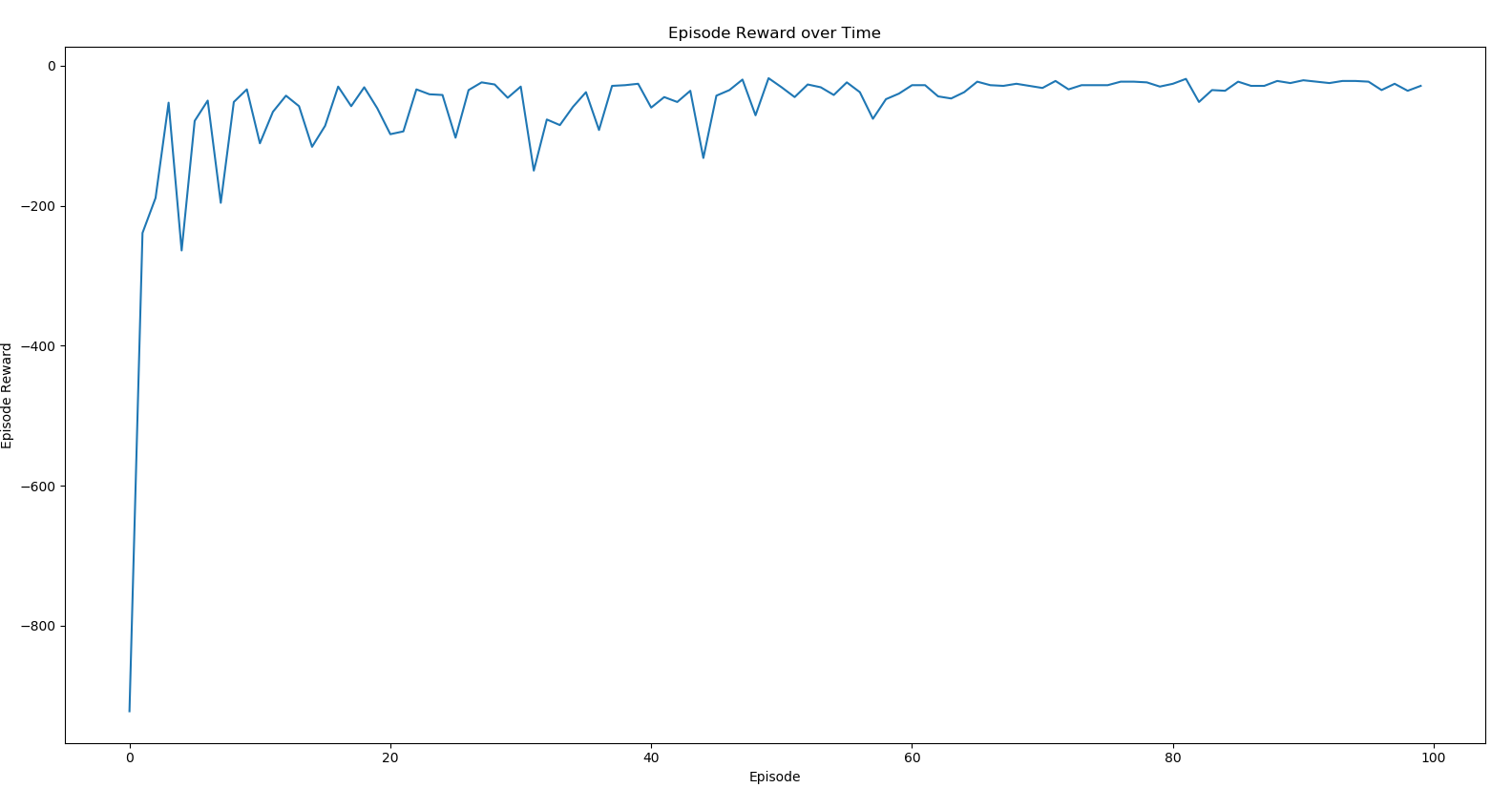
* **Alpha (α):** 0.5
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 50

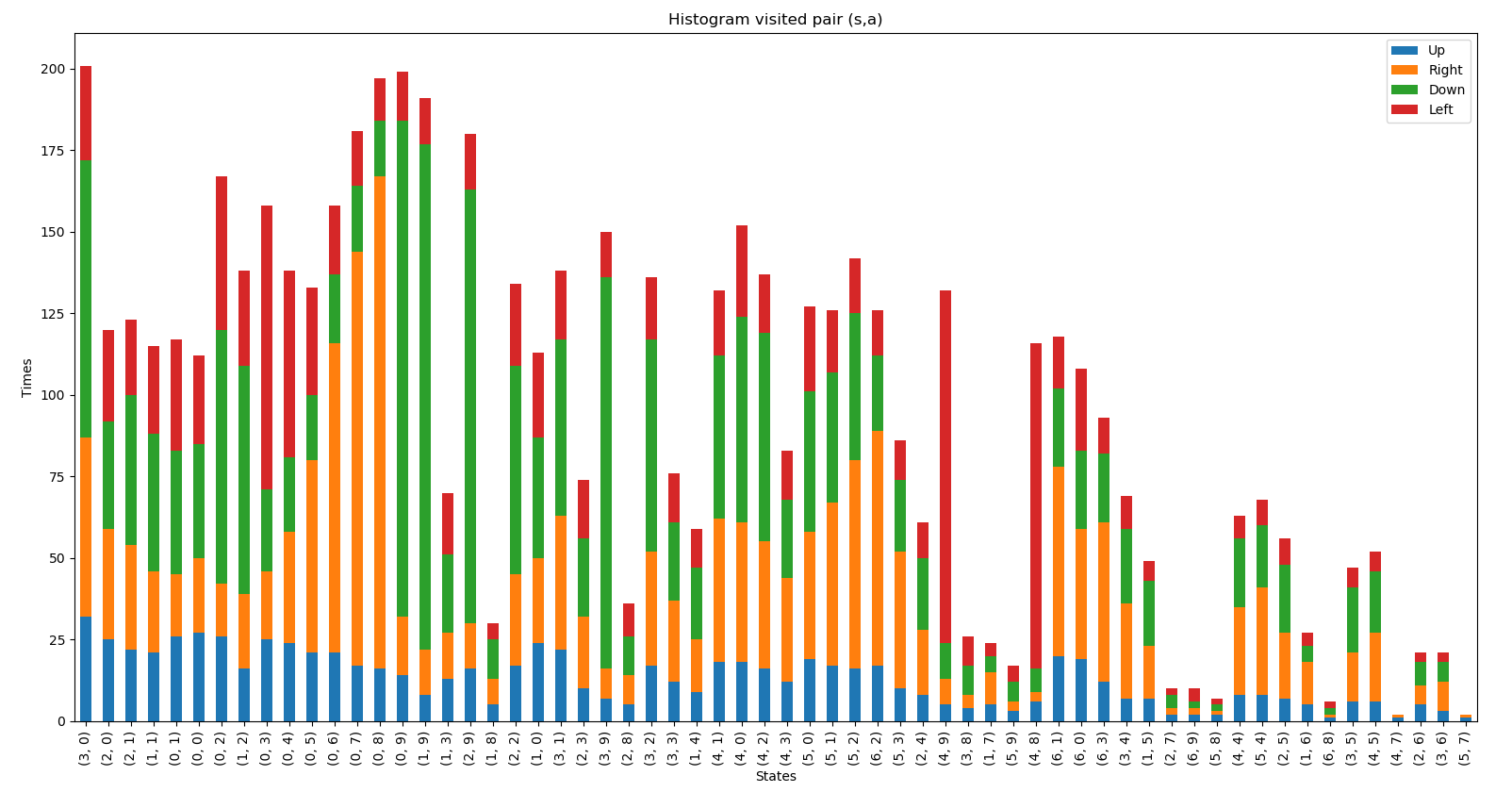


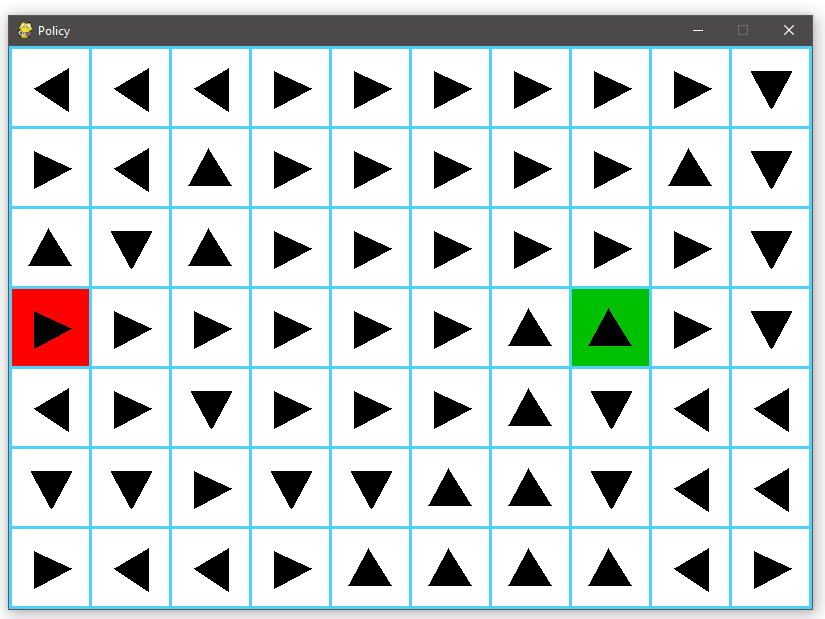


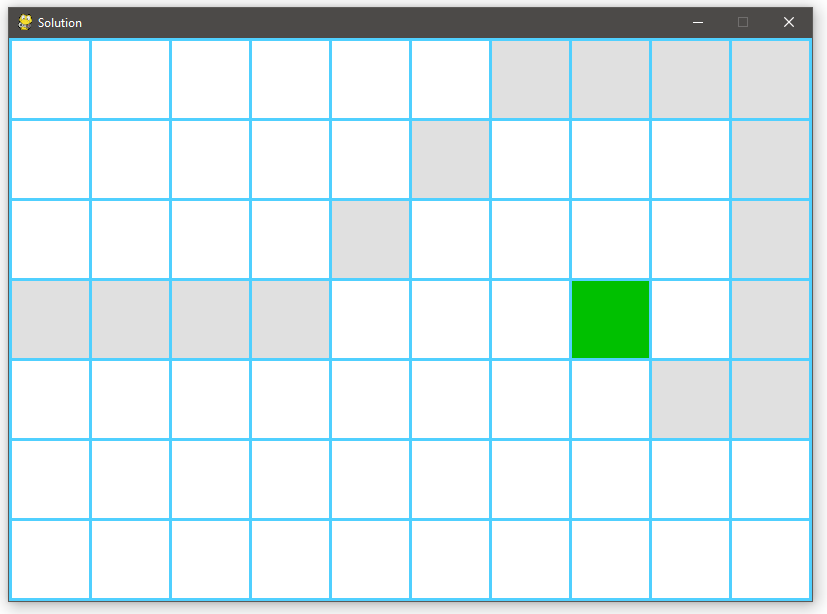
 **Alpha (α):** 0.5

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

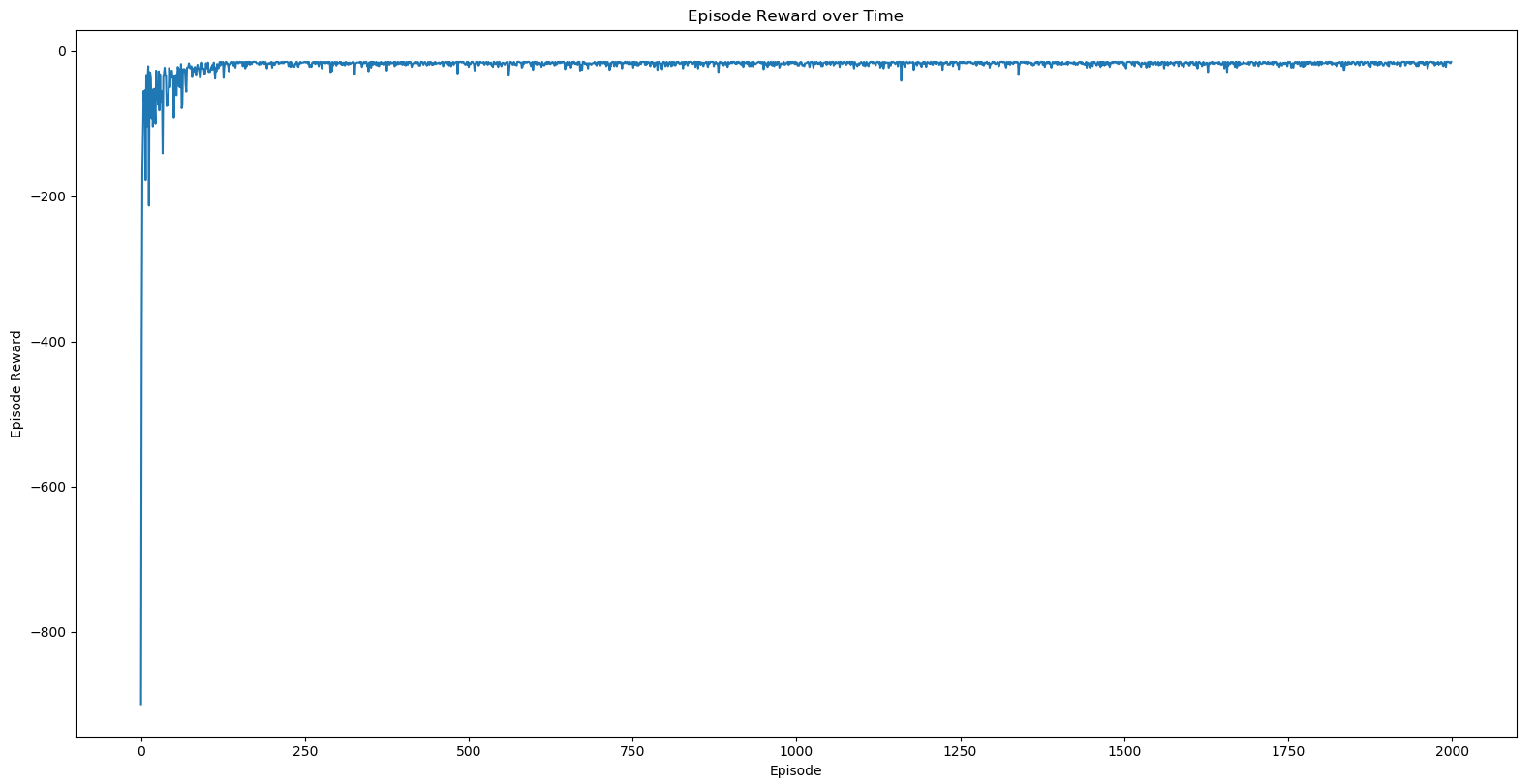


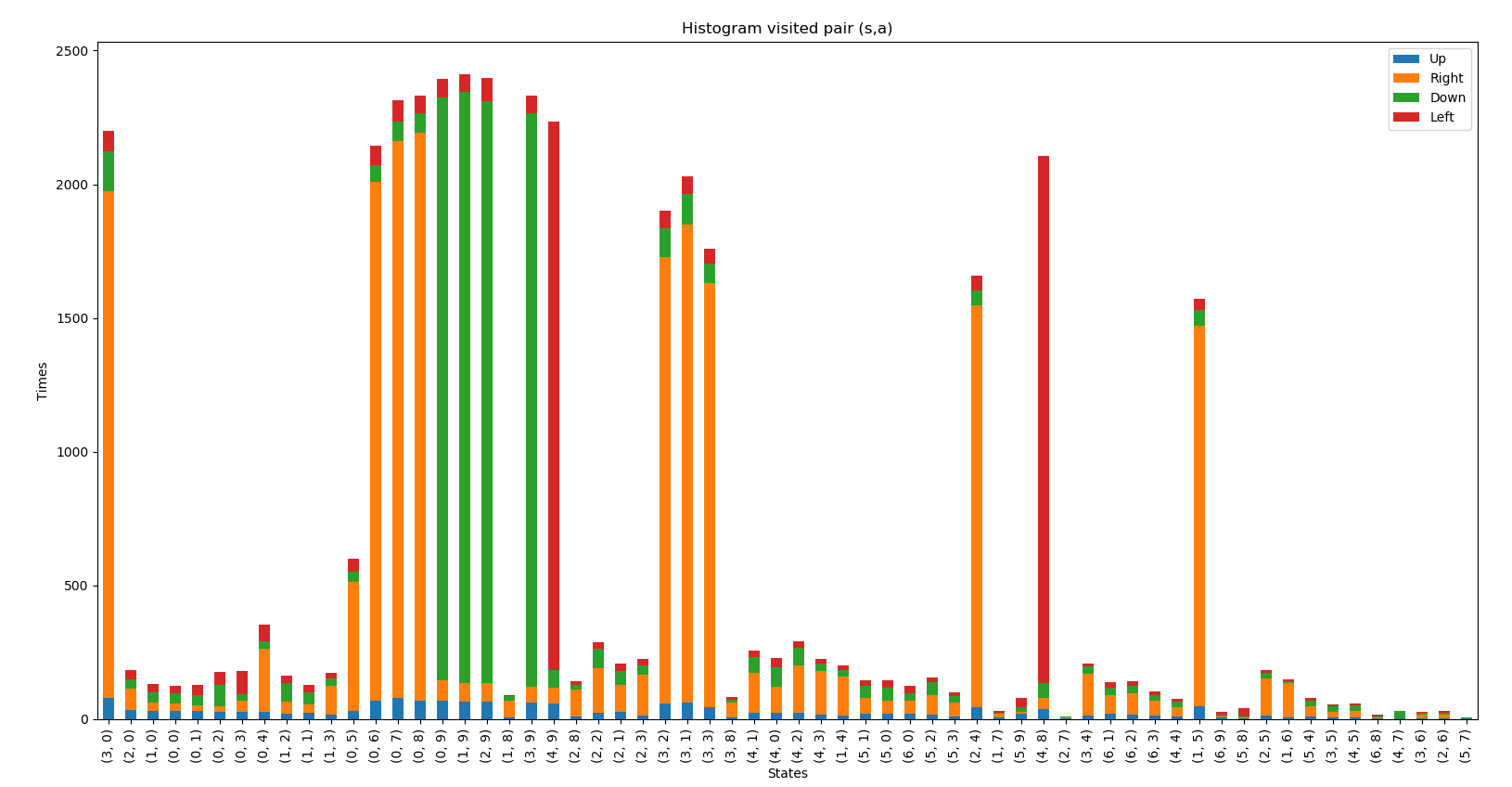


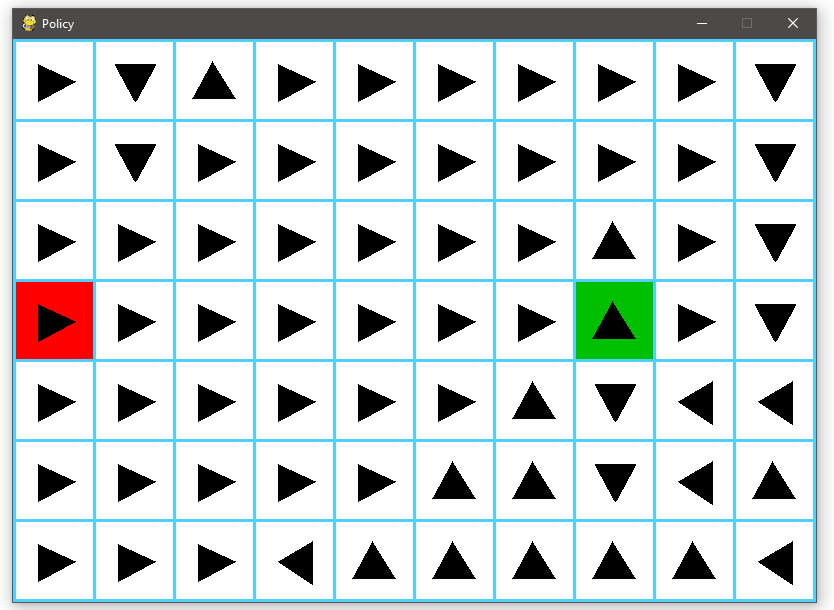


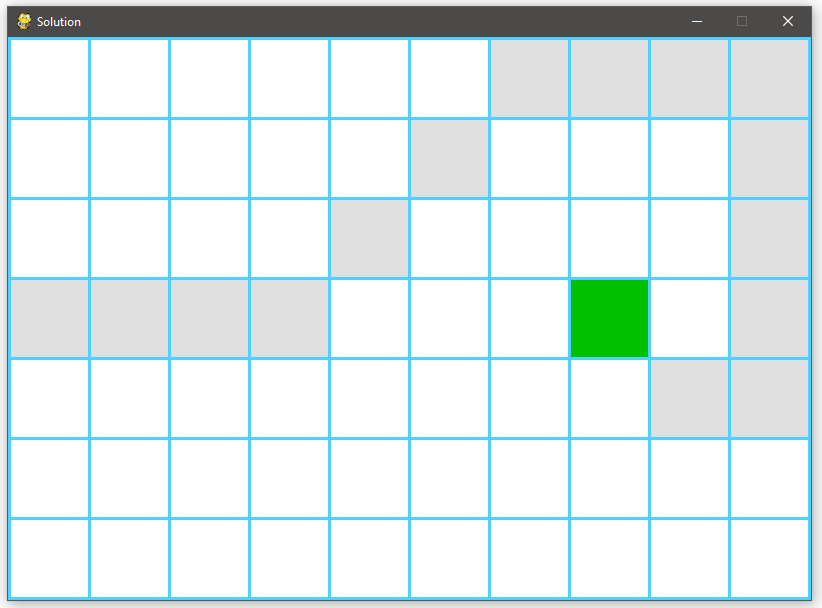


* **Alpha (α):** 0.5
* **Gamma (γ):** 1
* **Epsilon (ε):** 0.1
* **Number of Episodes:** 2000









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.