TBMI26 – Computer Assignment Reports Reinforcement Learning

Deadline – March 14 2021

Authors: Alexander Bois & Daniel Bissessar

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. Please upload the document in PDF format. **You will also need to upload all code in .m-file format**. We will correct the reports continuously so feel free to send them as soon as possible. If you meet the deadline you will have the lab part of the course reported in LADOK together with the exam. If not, you'll get the lab part reported during the re-exam period.

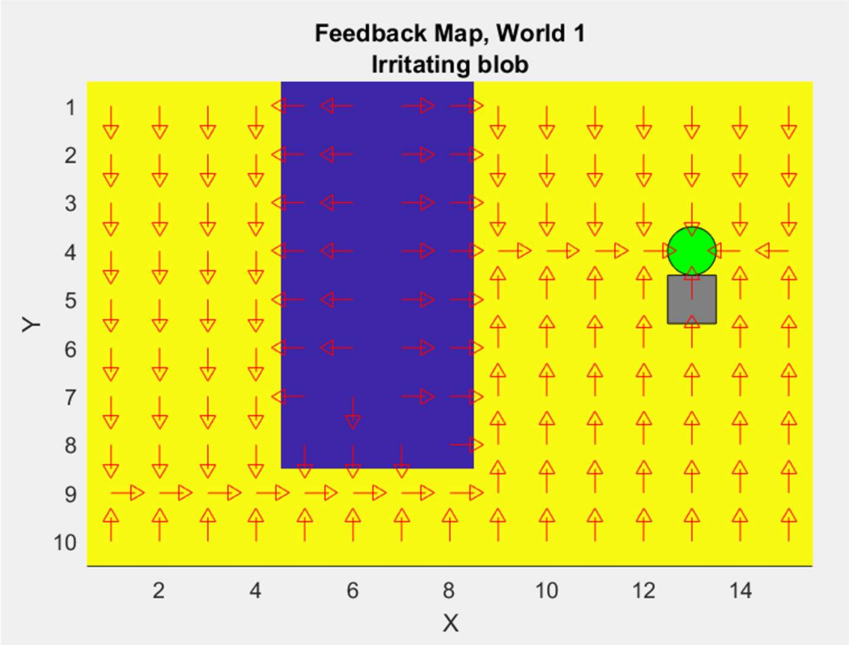
1. **Define the V- and Q-function given an optimal policy. Use equations and describe what they represent. (See lectures/classes)**
   The Q-function is defined as: $Q(s_k, a) = r(s_k, a) + \gamma V^*(s_{k+1})$
   Where the optimal F-function is defined as**:** $V^*(s) = max_a Q(s, a)$
   The Q function describes the different values for different actions in each and every state. While the value function is just the value of the best action to take in each state.

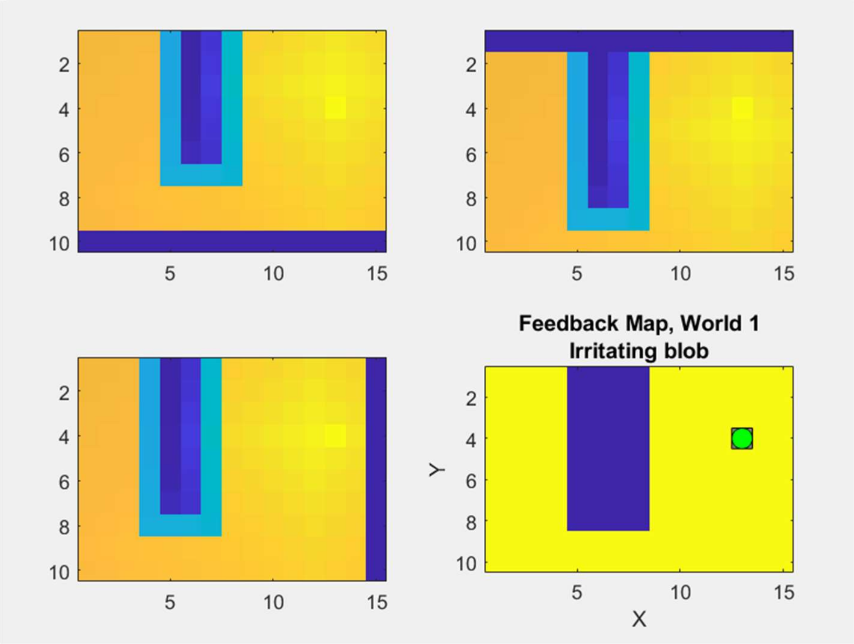
2. **Define a learning rule (equation) for the Q-function and describe how it works. (Theory, see lectures/classes)**
   The learning rule for the Q-function is defined as follows:
   $$Q(s_k, a_j) = (1 - \eta)Q(s_k, a_j) + \eta(r + \gamma V(s_{k+1}))$$
   To the left we have the updated Q for a state and an action. η is the learning rate, which controls how much influence previous estimate of Q and the actual reward plus value for taking action aj and ending up in state sk+1. r is the reward received for taking the action aj. Gamma is the discount rate controls the influence of close vs far rewards and how to prioritize these. V(sk+1) is the optimal value for the state that will be reached when doing action aj. The new Q value in stat sk for action aj is thus determined by the previous estimate of Q plus the reward for taking an action and the discounted value for ending up in the state sk+1.

3. **Briefly describe your implementation, especially how you hinder the robot from exiting through the borders of a world.**
   We hinder it from leaving the world by checking if the next_state is valid. If it is not, we update Q with a penalty. This is done according to rules. If the position is inside the map, we just update the Q as usual, but with the value function being based on the state that we came from. Is the position is outside of the map we update Q with minus inf for making such a move.

4. **Describe World 1. What is the goal of the reinforcement learning in this world? What parameters did you use to solve this world? Plot the policy and the V-function.**
   World 1 has an obstacle that is stationary. It is a rectangle approximately in the middle with a gap at the bottom allowing movement there. The blob starting position is random but the

goal is stationary. The goal is to create a policy that gets the blob to the goal in a short of a way as possible.



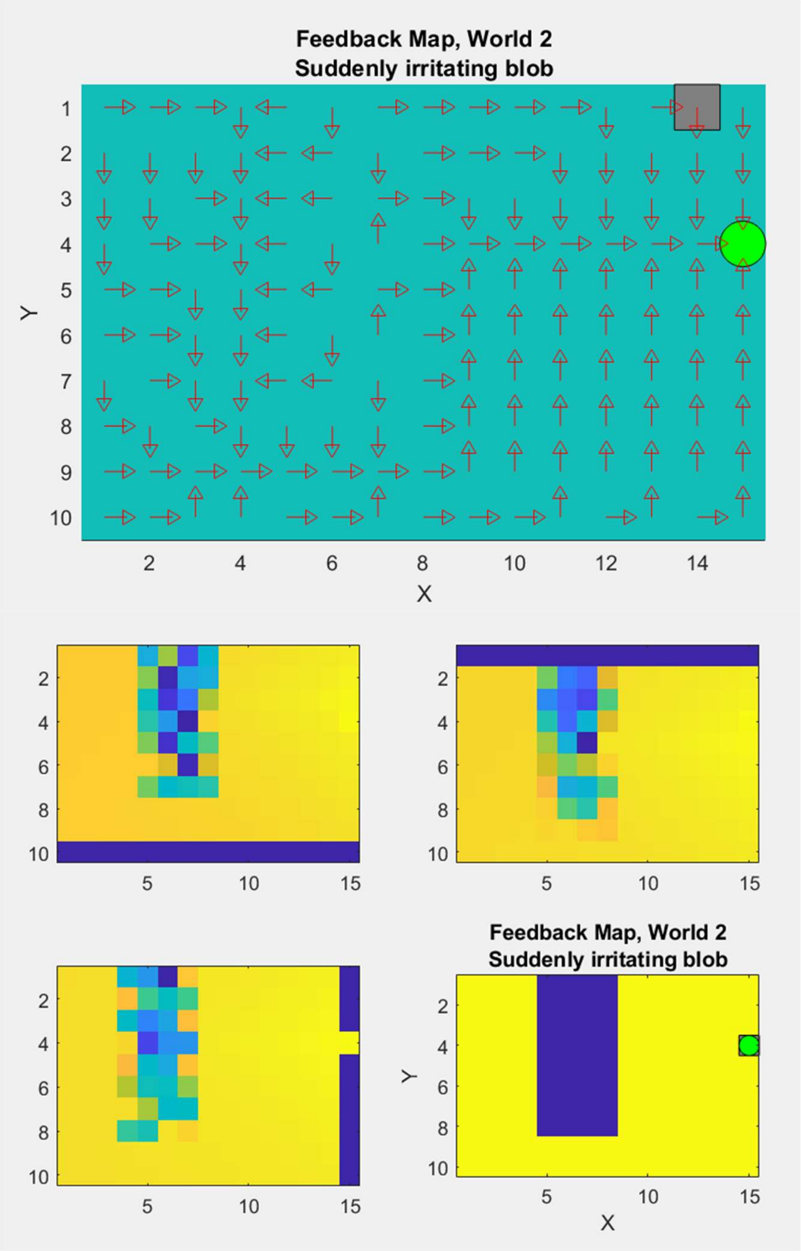


Eta: 1
Gamma: 0.9
Epsilon: 0.9 then to 0.3 when we have gone through 40% of the episodes
Episodes:1000

5. **Describe World 2. What is the goal of the reinforcement learning in this world? This world has a hidden trick. Describe the trick and why this can be solved with reinforcement**

learning. What parameters did you use to solve this world? Plot the policy and the V-function.

It is very similar to the previous world. But with another random variation where the obstacle is not always present. So it creates a policy where this possibility is handled by leaving this area as soon as possible and proceeding through the route we know is safe. This can be solved with reinforcement learning since it can learn to approximate the most optimal route according to the task over very many episodes since it can experience all of the possible scenarios.



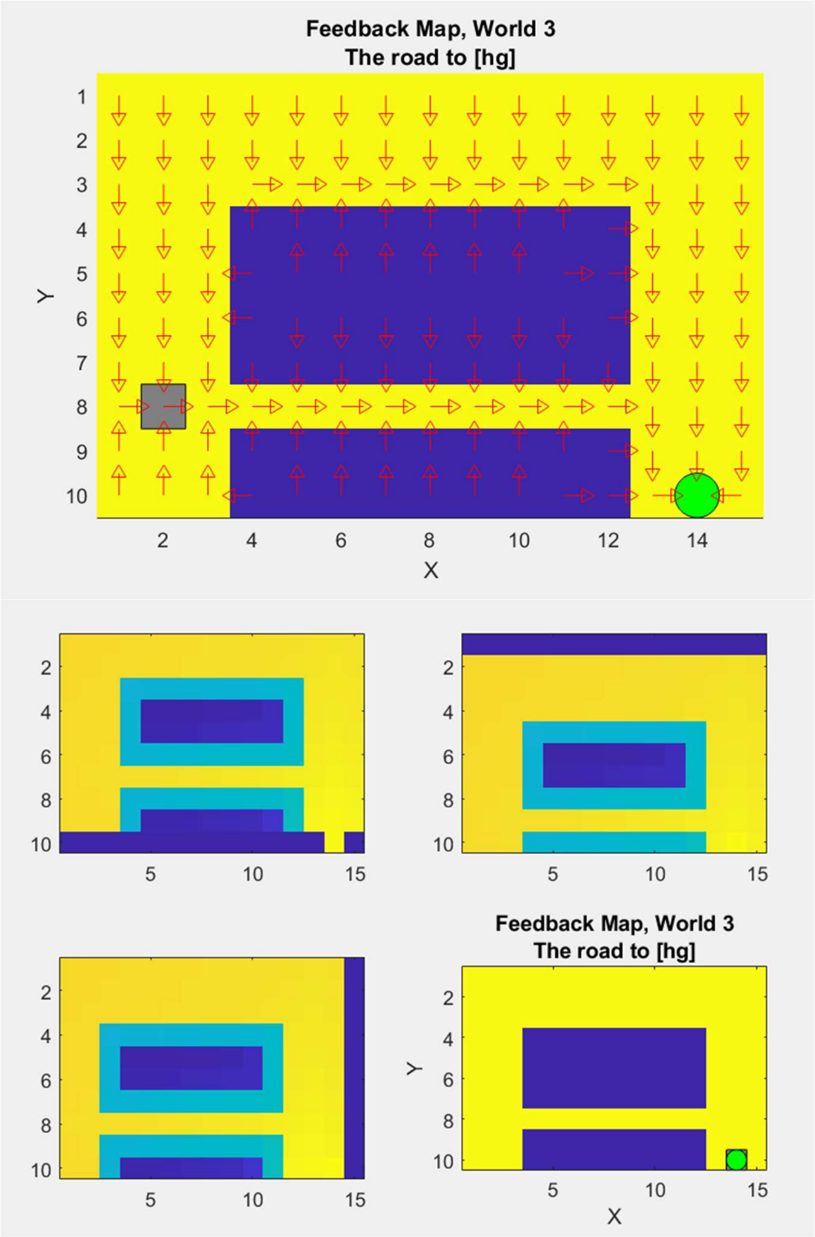
Eta: 0.3

Gamma: 0.9

Epsilon: 0.9 then to 0.3 when we have gone through 40% of the episodes

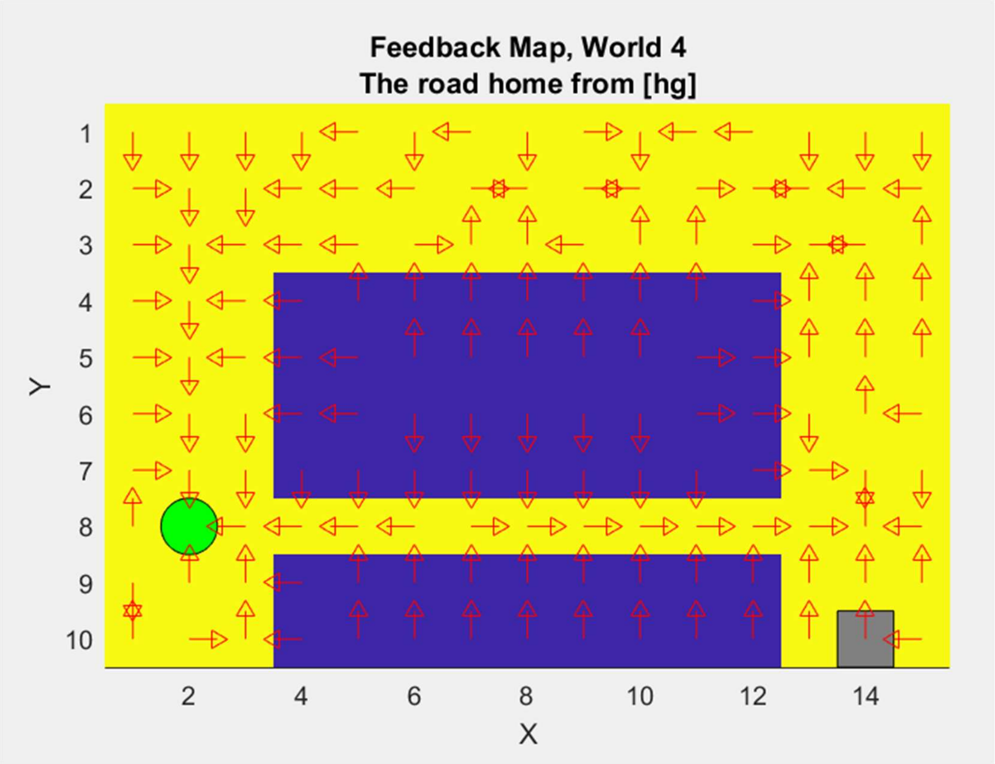
Episodes:1000

6. **Describe World 3. What is the goal of the reinforcement learning in this world? Is it possible to get a good policy from every state in this world, and if so how? What parameters did you use to solve this world? Plot the policy and the V-function.**
   It is a static world without no probability of leaving the trail and a static starting position. The goal in this task is thus to just find the fastest way from this point to the goal. With a focus on exploration we can achieve a good policy for every state.



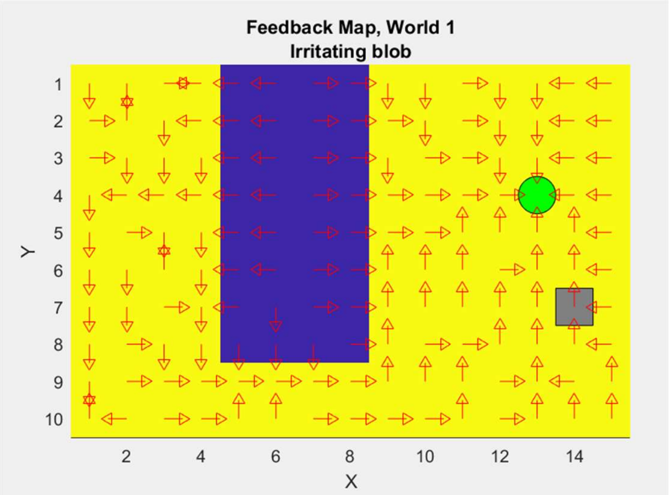
Eta: 1
Gamma: 0.9
Epsilon: 0.9 then to 0.3 when we have gone through 40% of the episodes
Episodes:1000

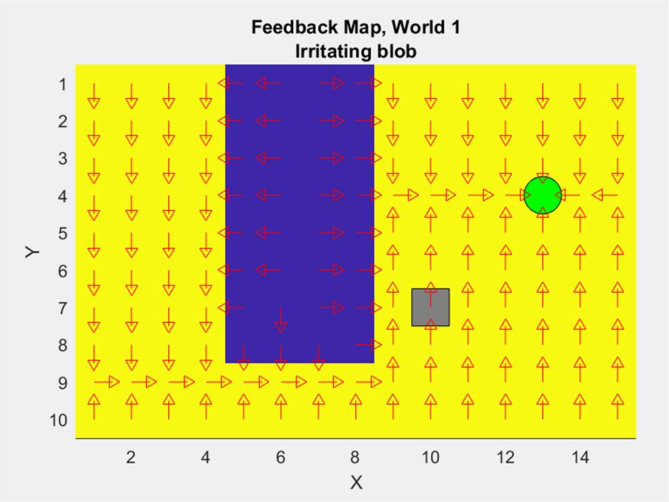
7. **Describe World 4. What is the goal of the reinforcement learning in this world? This world has a hidden trick. How is it different from world 3, and why can this be solved using reinforcement learning? What parameters did you use to solve this world? Plot the policy and the V-function.**
   This world adds the element of random movements in all direction with some probability. This makes it harder for the model to find its way, and the shortest route in the gap cant be used since it has a low value. The goal of this task is to reach the destination in such a manner that the risk of stepping outside of the yellow fields are minimized. This can be solved with reinforcement learning since it can learn to approximate the most optimal route according to the task over very many episodes.



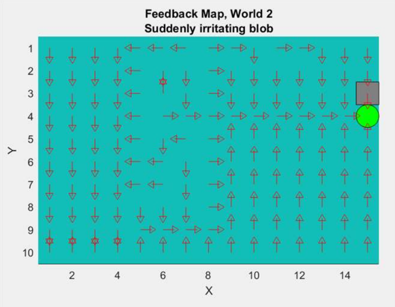
   Eta: 0.1
   Gamma: 0.9
   Epsilon: 0.9 then to 0.3 when we have gone through 40% of the episodes
   Episodes:50000

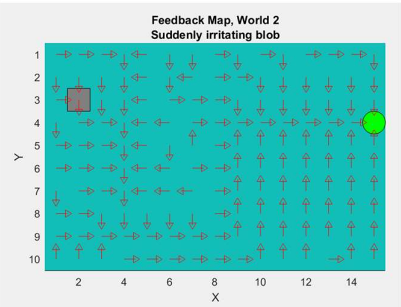
8. **Explain how the learning rate η influences the policy and V-function. Use figures to make your point.**
   The learning rate influences what value we place new vs old information when considering policy and value. In the pictures below we first have a low value for eta, and then a high. As can be seen the learning goes faster and converges faster with a higher learning rate.

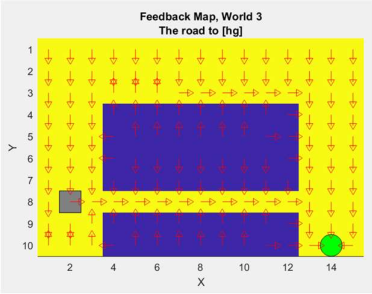


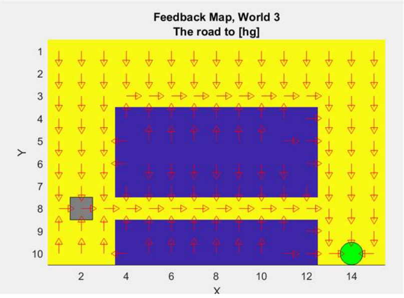


9. **Explain how the discount factor γ influences the policy and V-function. Use figures to make your point.**
The discount factor influences the policy by controlling how we value close vs far rewards. In the pictures below we first used a small gamma(pictures to the left) and then a large gamma(pictures to the right). As can be seen the when using the small it can get stuck in places due to it more valueing the close reward than the farther reward.

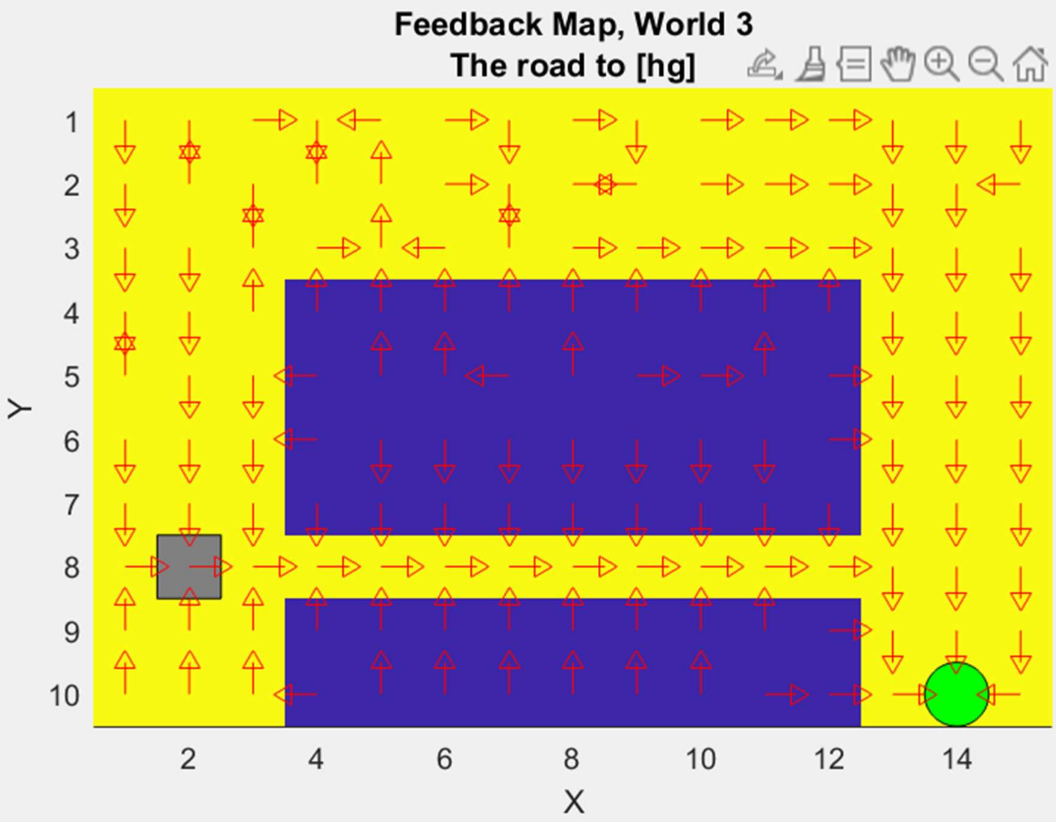


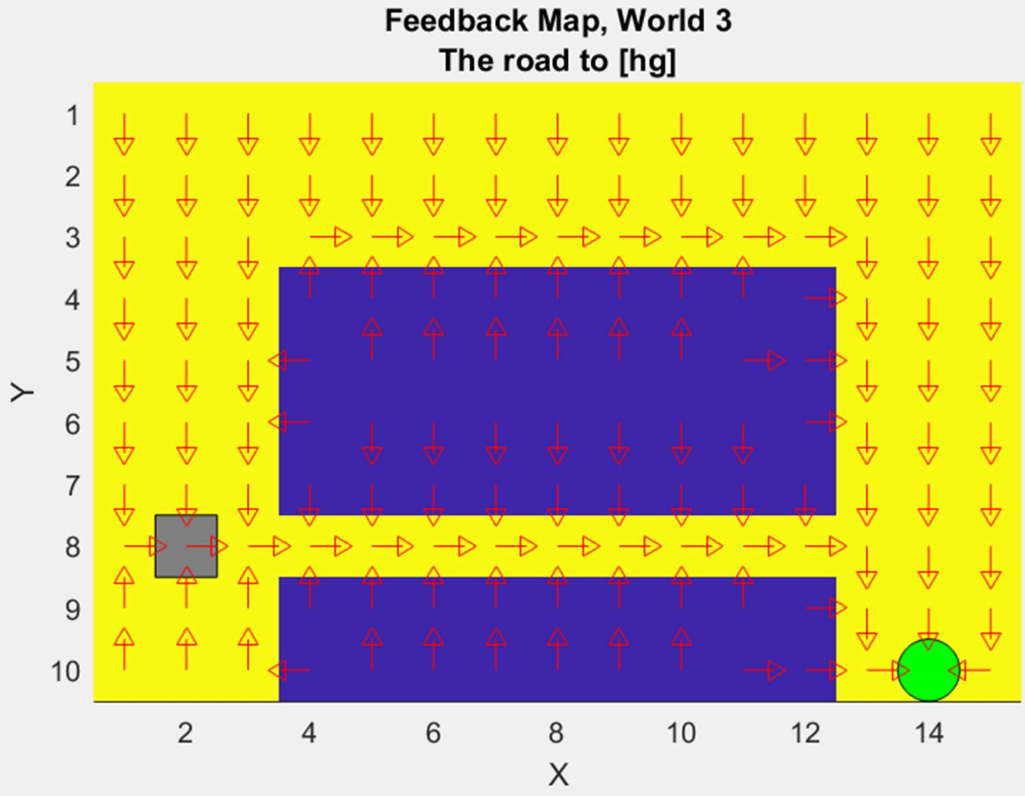






10. **Explain how the exploration rate ε influences the policy and V-function. Use figures to make your point. Did you use any strategy for changing ε during training?**
    The exploration rate controls to which degree the policy shall explore new policy vs exploit know policy to get more value. We used exploring strategy in the beginning, and after 40% of the episodes we choose a lower epsilon to make it more greedy/exploiting. As can be seen in the pictures where we used a greedy approach(small epsilon) in the first picture. It leaves the upper part of the map unexplored and with a very messy strategy. This is because it found the path in between the blocks and didn't explore the upper part that much. The second picture we explored a lot and we can observe that the strategy is defined for all parts of the map.


Feedback Map, World 3
The road to [hg]
Y
1
2
3
4
5
6
7
8
9
10
2
4
6
8
10
12
14
X


Feedback Map, World 3
The road to [hg]
Y
1
2
3
4
5
6
7
8
9
10
2
4
6
8
10
12
14
X

11. **What would happen if we instead of reinforcement learning were to use Dijkstra's cheapest path finding algorithm in the ''Suddenly irritating blob'' world? What about in the static ''Irritating blob'' world?**
    For the suddenly irritating blob we would have to run Dijkstra's algorithm for each new world we encounter. For the Irritating blob, which is static we could run Dijkstra's one time to find the optimal route and then use that every time.

12. **Can you think of any application where reinforcement learning could be of practical use? A hint is to use the Internet.**
    Some applications where RL could be used is e.g. self-driving cars or for a robot trader on the stock market.

13. **(Optional) Try your implementation in the other available worlds 5-12. Does it work in all of them, or did you encounter any problems, and in that case how would you solve them?**