Autonomous Agents
Report Assignment 3: Multi-Agent Planning and Learning

Authors:

Agnes van Belle *(10363130)*,
Maaike Fleuren *(10350470)*,
Norbert Heijne *(10357769)*,
Lydia Mennes *(10333843)*

October 20, 2012

1 Introduction

This report has been written for the Master Artificial Intelligence course Autonomous Agents. This report will contain the answers, motivation and explanation for our implementations of the tasks we had to accomplish in our third assignment for this course. These tasks were centered around the topic of 'Multi-Agent Planning and Learning'.

1.1 The environment

In all tasks there is assumed to be a grid world (of 11×11) with at least one predator and a prey in it. The agents can both move one tile forward each iteration. The direction they take (or if they move at all) is affected by probabilities (their policies). If they move over the edge of the grid they end up at the opposing side of the grid. We are focused on improving the decisions of all the agents, the predator(s) and the prey.

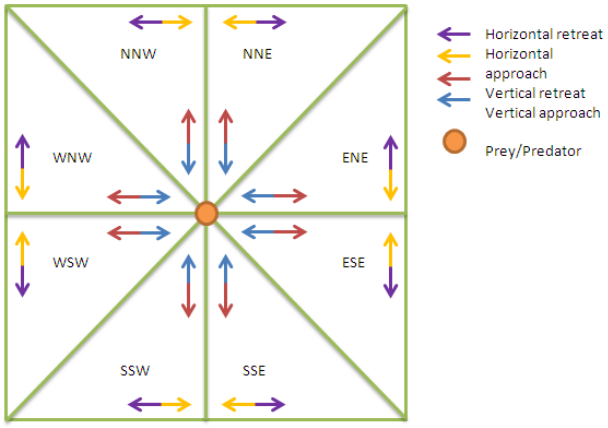
1.2 The state space representation

In this assignment, we used the efficient state space representation that we also used in the previous Assignment for the learning algorithms. To give a good understanding of our learning algorithms, which were built again using this efficient state space representation, we will once again explain how this representation works.

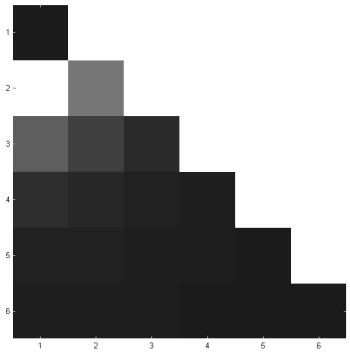
Figure 1(a) illustrates that there is a symmetry in the default state space, and thus that there were relatively much values redundantly computed. By using this symmetry in the default state space a much smaller state space was achieved.

Each state represents a distance between the prey and predator. These are represented in the lower left diagonal of a matrix, in which the x-axis is the relative horizontal distance in the MDP and the y-axis the relative vertical distance in the MDP. This matrix is shown in Figure 1(b). Combinations of positions of prey and predator for which the horizontal and vertical distances are equal are now treated equivalent. Also two combinations for which the horizontal distance in one equals the vertical distance in the other and vice versa are considered equal. In order to navigate through this state space different actions are required. These are: *horizontal retreat, horizontal approach, vertical retreat, vertical approach*, as illustrated in Figure 1(a), and of course the action *wait*. When interacting with the environment these actions are converted into corresponding actions in the real world. This only requires the relative direction of the prey (which is always

located at the centre, regardless of its coordinates) with respect to the predator. This is computed by using the difference in location of the prey and predator on the x- and y-axis.



(a) The 11×11 grid divided into eight symmetric pieces, with the corresponding possible moves which are also symmetric.



(b) Colormap of V-values, the brighter the color the higher the corresponding V-value. The prey is always located on the (1, 1) coordinate in this state representation.

Figure 1: Illustration of the symmetry and corresponding values of the new state space representation

Note that this state representation works only for two agents (here we talked about one prey and one predator). For Independent Q-Learning (see Section 2), in which more agents will be used, we will require a more extensive state space. That state space is, however, still built upon this one.

1.3 Implementation details

This report will not be about our exact code and implementation details. However, a class diagram of our code is provided in Appendix A.

2 Independent Q-Learning

Q-Learning is an off-policy temporal difference control algorithm. Temporal difference methods can learn directly from raw experience without a model of the environment's dynamics. Furthermore, it updates estimates based in part on other learned estimates, without waiting for a final outcome [3, pp. 133].

While the distinguishing feature of on-policy methods is that they estimate the value of a policy while using it for control, these two functions are separated in off-policy methods. The behavior policy, used to generate behavior, and the estimation policy, that is evaluated and improved, may in fact be unrelated. This separation is an advantage because the estimation policy may be deterministic (e.g., greedy), while the behavior policy can continue to sample all possible actions [3, pp. 126]. Its simplest form, *one-step Q-learning*, is defined by [3, pp. 148]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

For this assignment, we implemented Q-learning and used it for all our agents, the predators and the prey. Each agent will have its own state representation and learning and will view the other agents as part of the environment. We used ϵ-greedy action selection, which behaves greedily most of the time, but with probability ϵ, instead select an action at random, uniformly, independently of the action-value estimates [3, pp. 28]. In this case, we used $\epsilon = 0.1$. We initiated the values of our Q-learning table optimistically with a value of 15 for all cells in the table.

2.1 State space

The state space is of crucial importance to Independent Q-Learning. In the experiments described in the previous report, were we had just one prey and one predator, we initially used a state space that was an intuitive, yet cumbersome representation. We then changed the state space representation to a more efficient one in the second assignment, referred to as the 'efficient' state space, which led to a reduction of 697 times less states, resulting in just 21 different states. See Section 1.2 for details.

The default amount of states for this assignment would be 121^{p+1} for each agent, where p equals the number of predators. By using the efficient state space this has been reduced to $21 \cdot 121^{p-1}$ because the prey's position is fixed in the state space representation and the predator to which this state space belongs moves along those 21 states, the rest is used to determine in which state the predator currently resides. In this assignment however the prey also learns, so to make use of our state space representation, a single predator will be used as reference point every time so that the predator is fixed at the position $(0, 0)$ in our state space and the prey would be moving around the 21 states just like a predator.

The reduction is increased further by pointing to the same state for each combination of current positions of the other predators, though this is only effective for more than two predators. A simple example would be a case three predators and a prey, the current predator (p_1) which has to find his current state would take the position of the first other predator (p_2) and then the position of the second other predator (p_3) to locate his state in the state space. In this case it shouldn't matter which position, p_2 or p_3, would be looked up first. In this case both p_2 then p_3 and p_3 then p_2 would point to the same state. This reduces the state space to:

$$\text{stateSpaces}(p) = \begin{cases} 21 & p = 1 \\ 21 \cdot 121 & p = 2 \\ \frac{21 \cdot 121^{p-1}}{(p-1)!} & p > 2 \end{cases}$$

2.2 Implementation

Hier iets over met 4 predators lukte niet met 11x11 grid

2.3 Results

Bla bla

2.3.1 Using an 11 × 11 grid

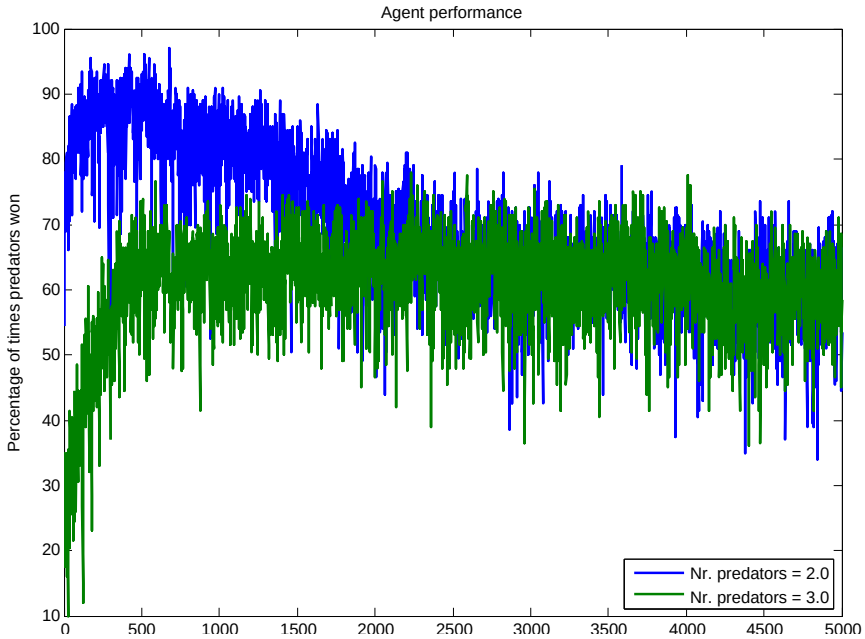


Figure 2: Bla

2.3.2 Using a 9 × 9 grid

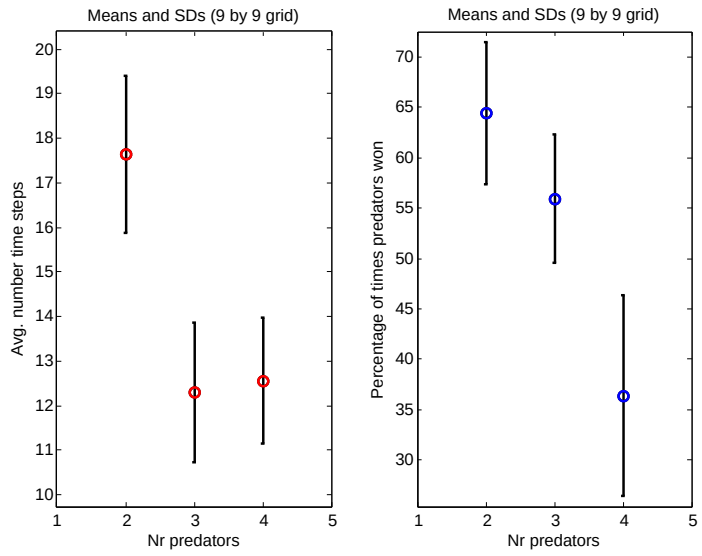


Figure 3: Error bar plot of performance in a 9×9 grid

3 (S) Minimax-Q algorithm

The Minimax-Q algorithm can be used for planning in a Markov Game. It is an implementation of the general Minimax principle for Markov Games such as the prey-predator scenario in this assignment. This algorithm needs a strictly competitive scenario and is therefore implemented for the one prey and one predator setting.

3.1 Minimax principle and translation to Markov Games

The Minimax principle is stated in **??** as: "Behave so as to maximize your reward in the worst case". This means that the policy should be such that the agent takes the action resulting in the highest expected reward under the assumption that the opponent will also take the action that results in the highest expected reward for the opponent.

The value of a state $s \in S$ in a Markov Game is:

$$V(s) = \arg\max_{\pi \in PD(A)} \arg\min_{o \in O} \sum_{a \in A} Q(s, a, o)\pi_a$$

where

$$Q(s, a, o) = R(s, a, o) + \gamma \sum_{s'} T(s, a, o, s')V(s')$$

Choosing the action that maximizes the reward in the worst case is than the policy that is guaranteed to result in an expected score of V no matter which action the opponent chooses, for the maximal value of V. This can be written down in constraints where there is a constraint for each action the opponent can take:

$$\begin{aligned} \pi_1 \cdot Q(s, a_1, o_1) + ... + \pi_n \cdot Q(s, a_n, o_1) &\geq V \\ \pi_1 \cdot Q(s, a_1, o_2) + ... + \pi_n \cdot Q(s, a_n, o_2) &\geq V \\ &... \\ \pi_1 \cdot Q(s, a_1, o_n) + ... + \pi_n \cdot Q(s, a_n, o_n) &\geq V \end{aligned}$$

Off course the probabilities of π are bounded by the constraints that they cannot be negative and they need to add up to one. After adding those constraints:

$$\begin{aligned} \pi_1 + ... + \pi_n &= 1 \\ \pi_1 &\geq 0 \\ &... \\ \pi_n &\geq 0 \end{aligned}$$

the maximal value for V can be found for which all constraints hold using linear programming. The resulting values for π for each state are the policy used by the agent.

3.2 Implementation

The Minimax-Q algorithm is implemented for this assignment using a JAVA-library for Linear Programming named Commons Math which can be found on **??**. The agent keeps track of a table of V-values to be able to calculate $Q(s, a, o)$ in order to find the right values for π. This can be done a number of times (or sweeps as they are called) as in Value Iteration. The resulting policy is used to behave in the environment to either catch the prey or avoid the predator.

3.3 Results

Three "matches" have been held using the Minimax-Q algorithm. The first match is a Random Prey versus a Minimax Predator, the second one a Random Predator versus a Minimax Prey and finally a Minimax Prey versus a Minimax Predator. After each sweep of planning 25 runs in the actual environment are held and the number of steps necessary to catch the prey is recorded. This averaged can then be compared for each sweep and between the different matches. The results for the first five sweeps can be found in 1, the pattern found here does not change when increasing the number of sweeps. The V-values for the Minimax Predator can be found in 2 and for the Minimax Prey in 3 . The resulting policies can be found in XXX.

	1	2	3	4	5
Minimax Predator vs Random Prey	9923	9432	9517	9545	9572
Random Predator vs Minimax Prey	282	5000000	5000000	5000000	5000000
Minimax Predator vs Minimax Prey	221	5000000	5000000	5000000	5000000

Table 1: Number of steps after each sweep for the three matches

0,0000					
10,0000	4,3644				
4,3644	1,7596	0,6019			
1,7596	0,6019	0,1624	0,0328		
0,6019	0,1624	0,0328	0,0047	0,0005	
0,0968	0,0328	0,0047	0,0005	0,0000	0,0000

Table 2: V-values of Minimax Predator

0,0000					
0,0000	0,0000				
0,0000	0,0000	0,0000			
0,0000	0,0000	0,0000	0,0000		
0,0000	0,0000	0,0000	0,0000	0,0000	
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Table 3: V-values of Minimax Prey

Appendices

A Class diagram

Hier klassediagram

B Policies for Minimax-Q

B.1 Policy for predator

	0	1	2	3	4	5	6	7	8	9	10
0	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
1	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
2	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000
3	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000
4	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
5	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	Prey	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
6	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
7	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
8	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
9	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
10	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000

Table 4: Policy resulting from Minimax-Q for the predator

B.2 Policy for prey

	0	1	2	3	4	5	6	7	8	9	10
0	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
1	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000
2	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000
3	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000
4	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000
5	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	Pred	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000	U 0,000 R 0,000 D 1,000 L 0,000 W 0,000
6	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
7	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
8	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
9	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000
10	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 1,000 D 0,000 L 0,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 0,000 R 0,000 D 0,000 L 1,000 W 0,000	U 1,000 R 0,000 D 0,000 L 0,000 W 0,000

Table 5: Policy resulting from Minimax-Q for the prey

References

[1] M.L. Littman, *Markov games as a framework for multi-agent reinforcement learning*. In Proceedings of the Eleventh International Conference on Machine Learning, pp. 157-163 San Francisco, CA. Morgan Kaufmann.(1994).

[2] Commons Math: The Apache Commons Mathematics Library (2012), http://commons.apache.org/math/.

[3] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT press. (1998)