# Artificial Intelligence and Intelligent agents – Semester assignment 2

You are going to investigate reinforcement learning (RIL) by means of a self-created Python program. Copying others work is strictly forbidden. To carry out this work you are going to create a grid world of 10x10 as shown below. The dark spaces are non-navigable areas. There are two traps and one big prize to be found. The cost of moving around is -5 for all moves. It is possible to move in all four directions, but not beyond the limits of the grid world.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A |  |  |  |  |  |  |  |  | -500 |  |
| B |  |  |  |  |  |  |  |  |  |  |
| C |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  |  |  |  |  |  |  |  |
| E |  |  |  |  |  |  |  |  |  |  |
| F |  |  |  |  |  |  |  |  |  | +1000 |
| G |  |  |  |  |  |  |  |  |  |  |
| H |  |  |  |  |  |  |  |  |  |  |
| J |  |  |  |  |  |  | -100 |  |  |  |
| I |  |  |  |  |  |  |  |  |  |  |

1. In the beginning you are going to investigate and compare two methods of RIL, model-based and model-free learning. For the former you will apply a random policy to generate values and demonstrate policy iteration to find the optimal (executable) policy for the agent. The agent can go forward, backward, left, right with an equal probability (p=0,25). When your agent hits the wall it stays in its original state until next action.
2. In addition to this you are going to apply a Q-learning approach for the same purpose. Both approaches should be possible to demonstrate with your program. A basic visual element for this is needed. More advanced graphics is considered a plus and will be given some additional credit. For comparison you are going to record and show relevant statistics. Minimum is time to satisfactory convergence and establishment of an optimum, number of samples/episodes needed and evolvement of error rate. Show this in proper graphs.
3. Try to think of the different slates in the grid as waypoints. Show how you would use a DQN type of approach with experience replay to find an approximation and guide the exploration and improve the performance of the Q-learner.
4. The big reward is actually associated with a bounty for catching a bank robber. He has been hiding in G9 for a long time, but is now getting uneasy. He starts to move around. First he hops between his original hiding place and one other “safe house” in F2. He uses a mixed strategy. The percentage time spent in the new safe house is 35%. This means that for every step taken by your agent (bounty hunter) there is a probability that the bank robber could be in this place with a probability of p=0,35 instead of in D9. An additional action is added to the bounty hunter. In addition to the 4 previous actions he can wait. Expand one of the methods applied in a to catch the thief. Again, record the statistics.
5. The bank robber now feels very uneasy. He stashes away his loot. At the same time, he is applying more intelligence to avoid the bounty hunter. In fact, he is a RIL-learner himself. So, he tries to stay at arm’s length at all times. When both share the same state the bank robber is killed. However, the bank robber has another concern. He is afraid that his loot is going to be stolen by others. A woman that he trusts keeps moving the loot for him to new hiding places so that in addition to staying away from the bounty hunter he needs to track down the loot which is worth 1000. For the bounty hunter the bank robber is still worth 1200 with or without the loot. If the bank robber manages to pick up the loot his helper, the woman, will replace it somewhere else in the grid, but she can only place it one out of three hiding places: E7, A10, J5 and F3. the bounty hunter and the bank robber maintain their positions for the new hunt. When the bounty hunter catches the bank robber the game is over. Record the statistics.

# Final report

The final report should be in the form of a “scientific article” using two columns and an abstract. The IEEE format is recommended. It should be between 6 and 10 pages long, including images and reference list. It is important that you show graphs that offer an insight on how the agents learn and make transparent whether learning converges.

# Submission

Submission will be in the form of a report (see above) and Python code version. Information on the exact submission date and form will be given later in the course, but will be very late in October or early November.