Machine Learning
hugomcp@di.ubi.pt, 2019/20

Practical Project

Reinforcement Learning

1. Create a Pyhton script that generates random "N x N" boards, with each cell having only two possible types: "**Safe**" (penalty=1) or "**Dangerous**" (penalty=100).

2. Generate two random cell positions (each one provided in a (row, column) format), corresponding to the current position of the agent and to the final position. (represented in yellow and green color in the example below).

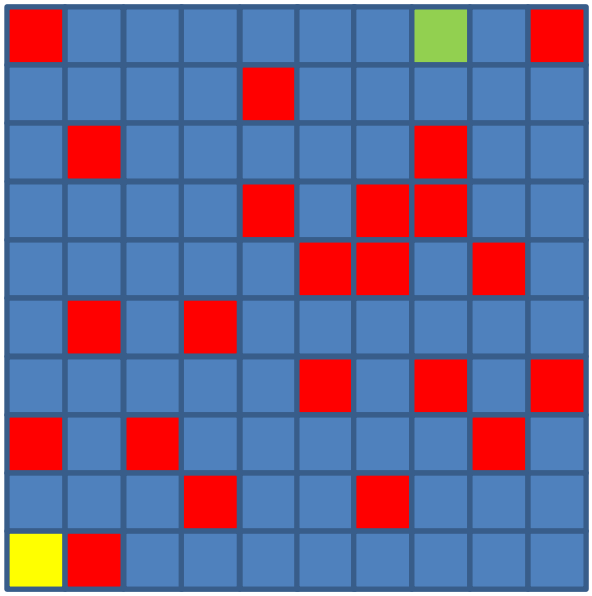

Fig: 1. Example of a 10x10 board. What is the best way to move from the yellow to the green cell?

3. Using Reinforcement Learning techniques (Q-learning algorithm), design and implement a solution for moving from the current state to the final position in as few movements as possible, avoiding the "**Dangerous"** cells as much as possible.

4. Compare different strategies for defining the "state" in terms of the spatial **computational cost** of the algorithm (the amount of storage required) and the **quality** of the **solutions** generated (which corresponds to the cost of the overall path, i.e., the sum of the penalties for the cells composing the best path).

5. Perceive the variations in the results, with respect to the α, γ values used in the Q-Table update formula.

6. Prepare a small report, describing your main findings.