# Artificial Intelligence: Programming 3 (P3)
# Reinforcement Learning

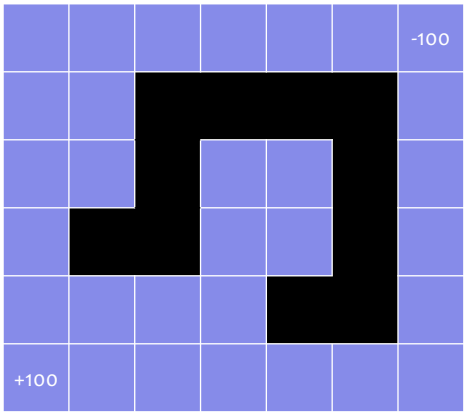
Instructor: Dr. Shengquan Wang

Due Time: 10PM, 4/4/2020

In this project, we aim to implement one of Reinforcement Learning algorithms: the `Q-Learning` algorithm.

## 1 Instructions

We extend the windy maze defined in P1 with probabilistic outcome after an action and two terminal states (left bottom and top right). It becomes a MDP problem. The maze map is shown in the following figure. However, we assume that the agent doesn't know either the reward function or the transition




model. The agent aims to run many trials in order to obtain Q-value for each (state, action) pair and the optimal action at each state.

**Environment** In your implementation, you need to simulate the windy maze environment: We assume that the wind comes from the south and the cost of one step for the agent is defined as follows: 1 for moving northward; 2 for moving westward or eastward; 3 for moving southward. The reward will be the negation of the reward. The agent can drift to the left or the right from the perspective of moving direction with probability 0.1. If the drifting direction is an obstacle, it will be bounced back to the original position. If the agent falls into any terminal state, it can't move out.

**Reinforcement Learning** In your implementation, you will generate many trials, each of which will result in a trajectory of (state, action, reward) tuple. The agent will use the $\epsilon$-`Greedy` algorithm to choose an action at each state along each trajectory, where $\epsilon = 0.05$: the agent chooses a latest optimal

action at each state with 95% and a random action with 5%. The initial state for each trial is chosen randomly and each trial will end at the goal state. Along each trajectory, the agent will use `Q-Learning` to update the Q-values. Since the reward function $R(s, a)$ here depends on both the state and the action taken at this state, the Q-value update equations should be revised accordingly.

$$N(s,a) \leftarrow N(s,a) + 1 \tag{1}$$

$$Q(s,a) \leftarrow Q(s,a) + \frac{1}{N_{s,a}} \Big( R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \Big) \tag{2}$$

We choose $\gamma = 0.9$.

**Testing and Outputs** In your testing, generate $20,000$ trials starting from a random open square. We initialize the Q-values for each trial at any state-action as 0 except for two terminal states with $+100$ and $-100$ respectively. After $20,000$ trials, report the following three outcomes for each algorithm:

- the access frequency at each state-action $N_{s,a}$;
- the Q-value function at each state-action $Q(s, a)$;
- the optimal action at each state-action.

The expected outcome should look like as follows:

- **Table of $N(s, a)$:**

```
    1519              2062              3029              5021              8371              44
48        50      98        90      3974      94      2382      100     1389      129     2013      29
    1531              4474              70                101               148               29
    114               80                                                                      55
85        99      5187      74      ####              ####              ####              ####      77        48
    6595              75                                                                      4390
    115               112                               5733              6770                        3405
129       123     2383      41      ####              112       133     1083      97      ####      697       1137
    8654              38                                1853              121                         2056
    131                                                 143               91                          563
202       185     ####              ####              74        54      1385      35      ####      234       281
    9680                                                4000              26                          2612
    227               140               83                74                                        185
221       221     9542      137     6461      81      4655      65      ####              ####      116       104
  17401               122               92                71                                        4034
                      338               846               118               120               125               99
                  7462      105     6616      88      6712      75      6103      77      5195      76      4340      63
                      221               89                104               72                70                70
```

- **Table of $Q(s, a)$:**

```
    -5.2            -5.3            -5.7            -6.0            -6.5            -9.7
3.8     -1.6    6.9     0.8     8.1     -3.6    1.6     -6.5    -2.9    -8.0    -7.3   -77.7       -100
    24.7            17.0            -2.1            -6.0            -8.1           -11.7
    13.7            9.2                                                                    -69.1
24.7    18.4    29.1    18.6        ####            ####            ####            ####        -14.5  -20.3
    40.3            28.4                                                                   -12.2
    32.6            8.4                             -6.0            -5.9                   -10.7
44.3    34.5    42.2    27.0        ####        -3.0    -4.8    8.3     -6.0        ####        -10.7  -10.7
    54.5            30.8                            17.4            -3.9                   -10.1
    48.5                                            -0.3            -3.7                    -9.2
43.2    49.2        ####            ####        18.2    10.8    18.3     1.5        ####        -9.1    -9.0
    68.8                                            31.4            5.4                    -5.4
    61.8            56.3            46.8            21.3                                   -7.0
72.0    64.3    71.1    53.0    59.3    39.3    46.0    32.5        ####            ####        -5.5    -5.4
    82.8            67.6            52.2            37.7                                   1.2
                    60.2            45.5            33.3            20.7           7.0     -1.4
    +100        83.7    61.6    69.7    43.2    53.7    28.8    37.4    15.9    22.9     5.4    10.5     1.7
                    55.7            55.5            38.9            25.9           11.0    4.2
```

- **Table of the optimal action at each state:**

```
vvvv    vvvv    <<<<    <<<<    <<<<    <<<<    -100
vvvv    <<<<    ####    ####    ####    ####    vvvv
vvvv    <<<<    ####    vvvv    <<<<    ####    vvvv
vvvv    ####    ####    vvvv    <<<<    ####    vvvv
vvvv    <<<<    <<<<    <<<<    ####    ####    vvvv
+100    <<<<    <<<<    <<<<    <<<<    <<<<    <<<<
```

  where `<<<<`: moving westward; `^^^^`: moving northward; `>>>>`: moving eastward; `VVVV`: moving southward; +100 and −100: the terminal rewards.

# 2 Submission

Form a group on Canvas if you want to work with another student. You are going to report the following things:

(a) Describe in details how you implemented the following modules in the report: `environment simulation`, $\epsilon$-`greedy`, and `Q-learning update`.

(b) Comment your code in details so that the grader can understand it well.

(c) Include the screenshots of all above testing outcomes. Each screenshot should include your username and the current time, which show that you did it by yourself.

(d) Specify the contribution made by each member if you work as a group.

The report should be written in a ".docx", ".doc", or ".pdf" format. Submit both the report and the source code to the assignment folder P3 on Canvas. Any compression file format such as `.zip` is not permitted.