Collision avoidance and GridWorld management

- If two agents attempt to move to the same position, one is randomly prioritized and moves success- fully, while the other receives the collision penalty (-0.05) and their position is unchanged.
- At the beginning of each episode, agents spawn at their respective location and each agent simultaneously takes an action until all agents have reached a target or 31 steps have elapsed.
- Once a target has been reached by one agent, they occupy that position and stop interacting with the environment until the episode terminates.

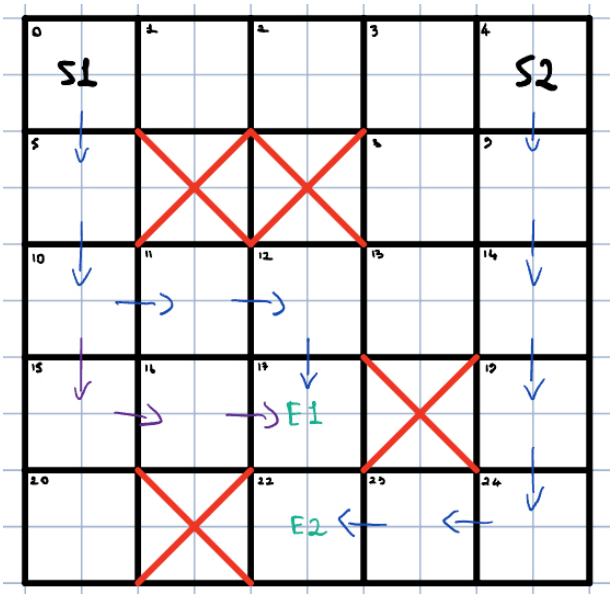
FAQ – Learning settings and results / 2 agents on a grid world 5x5

- M = 200 (epochs)
- k = 7 (episode's length)
- beta = 0.5
- eps = $\left(1 - \frac{m}{M}\right)^2$
- x_i = ε - greedy probability structure
- $Q[s,a] = Q[s,a] + \min\left(\frac{\beta}{x_i}, 1\right) * (reward + gamma * \max\left(Q\left[s_{prime},:\right] - Q[s,a]\right)$
- When collision happens between the two robots, we randomly choose one of the two that executes the chosen action, the other one avoids the crush and remains in the current state, obtaining a negative reward of -1.
  In the case one of the two robots occupies a target position and the other one tries to reach the same one, it will be the chosen one that will receive the negative reward.
- The episode ends when both robots have reached the target or when both robots have done 7 actions.
  When a robot reaches the target before the other one, should it get a smaller reward than when reached together?
- RESULTS:

```
Q1 matrix updated:
[[ 0.          0.3122      0.          0.06284881]
 [ 0.         -0.9399055   0.18096156 -0.1       ]
 [ 0.         -1.         -0.01191602 -0.1003198 ]
 [ 0.         -0.07741153 -0.09951363 -0.08924391]
 [ 0.         -0.05314061 -0.05314061  0.        ]
 [ 0.18098     0.458       0.         -0.5878    ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.3122      0.62        0.          0.62      ]
 [-0.29142445  0.8         0.          0.35453398]
 [ 0.          0.75251888  0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.458       0.458       0.          0.8       ]
 [ 0.62       -0.1         0.62        1.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.62        0.          0.         -0.45994014]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]]
```

```
Q2 matrix updated:
[[ 0.         -0.09200537  0.         -0.09375   ]
 [ 0.         -1.         -0.1        -0.09769894]
 [ 0.         -1.         -0.1        -0.09214337]
 [ 0.         -0.19       -0.1         0.06284907]
 [ 0.          0.18098    -0.04352476  0.        ]
 [ 0.          0.          0.         -0.61728395]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [-0.04477602 -0.10937211 -1.04346343 -0.1       ]
 [ 0.06038382  0.3122     -0.14147791  0.        ]
 [ 0.          0.          0.         -0.05425347]
 [-0.5        -0.09255403 -0.05425347  0.        ]
 [-0.5        -0.22040591 -0.09659373 -0.12810928]
 [-0.18646451 -0.99949967 -0.10619189  0.29272225]
 [ 0.16343396  0.458       0.14592004  0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [-0.76207895  0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.3122      0.62       -0.442       0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [-0.1         0.          1.          0.62      ]
 [ 0.458       0.          0.8         0.        ]]
```

- OPTIMAL POLICIES OBTAINED:



- Last Collision: epochs 54
- Epochs with the great number of collisions: 21 with 2 collisions
- Total Collisions: 10/200 (5%)

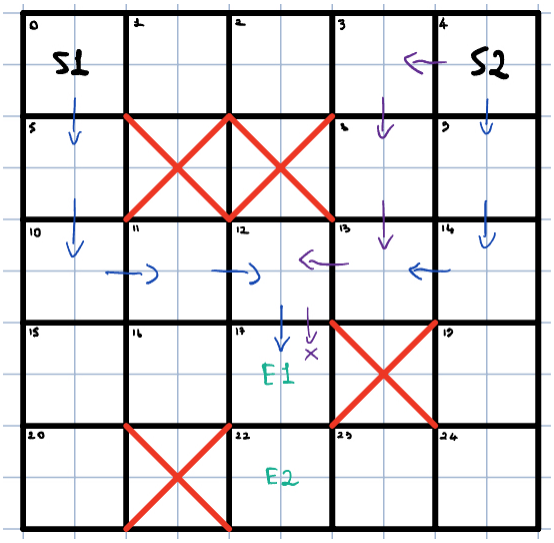
Q–Learning Settings and Results / 2 agents on a grid world 5x5

- M = 200 (epochs)
- k = 7 (episode's length)
- alpha = $\left(1-\frac{m}{M}\right)$
- eps = $\left(1-\frac{m}{M}\right)^2$
- $Q[s,a] = Q[s,a] + \text{alpha} * (reward + gamma * \max(Q[s_{prime},:] - Q[s,a])$
- When collision happens between the two robots, the episode ends.
- The episode ends when both robots have reached the target or when both robots have done 10 actions.
- RESULTS:

```
Q1 matrix updated:
[[ 0.          0.3122      0.          0.0625857 ]
 [ 0.         -0.87988943  0.1809435  -0.1       ]
 [ 0.         -0.99999949 -0.09999999 -0.09999999]
 [ 0.         -0.0999935  -0.0895875  -0.0931575 ]
 [ 0.         -0.0885     -0.0405     0.         ]
 [ 0.18042674  0.458       0.         -0.58780318]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.         -0.71        0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.3119298   0.31162132  0.          0.62      ]
 [-0.29396086 -0.08854619  0.41779866  0.8       ]
 [-0.05250001  1.          0.56873366 -0.0950932 ]
 [-0.071      -0.53        0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.45795184 -0.0998862   0.         -0.09999986]
 [-0.092      -0.675      -0.099925    0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.12284082  0.          0.         -0.901825  ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]]
```

```
Q2 matrix updated:
[[ 0.         -0.077       0.         -0.0765    ]
 [ 0.         -0.9984965  -0.094595   -0.09241   ]
 [ 0.         -0.9998877  -0.09999777 -0.1       ]
 [ 0.          0.458      -0.1         0.14372184]
 [ 0.          0.3122      0.3122      0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.31143121  0.62       -0.45888203  0.30803198]
 [ 0.18097984  0.458       0.45786444  0.        ]
 [ 0.          0.          0.          0.        ]
 [-0.866775   -0.0335      0.          0.        ]
 [-0.09999815  1.         -0.09150463  0.61943513]
 [ 0.45799718 -0.28000176  0.8         0.45799389]
 [ 0.31214008  0.31182333  0.62        0.        ]
 [ 0.          0.          0.          0.        ]
 [-0.0335      0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.45788797 -0.09999417 -0.62221724  0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.        ]
 [-0.25772     0.          0.9008      0.        ]
 [-0.09916     0.          0.3322      0.        ]]
```

- OPTIMAL POLICIES OBTAINED:



- Total Collisions: 115/200
- Observations: with the same number of epochs and the same episode length, the simple q-learning demonstrate to not avoid the collisions and is not able to coordinate the two agents to reach heterogeneous targets. In fact, they both learn how to reach E1, that will always cause a collision.