1 Q-Learning

The Markov Decision Process has to be modeled like this:
$M =< S, A, P, R, \gamma >$
$S : s_{1,1}...s_{3,3}$
$A : U \in \{N, E, W, S\}$
$P : P(Transition(S_{i,j} \times A \times S'|S) = (\sum u_{s_{i,i}})^{-1}$
$R_{a(S,S')} : S \times U \text{ with } r = \begin{cases} 0, \text{if Q(0,u)} \\ -1, \text{else} \end{cases}$
$\gamma : 0.5$

1.1 Update Rule

With costs c(i,u) as negative reward (state S=i and S'=j) the Q-Learning Update Rule can be expressed using α as learning rate as follows:
$Q_{k+1}(i, u) := (1 - \alpha) \cdot Q_k(i, u) + \alpha(c(i, u) + \gamma \min_{u' in U} Q_k(j, u'))$

1.1.1 Handling Goal State

As shown in the MDP the absorbing terminal state $Q(0, u) = 0| \; \forall u \in U$
$Q_{k+1}(i, u) := (1 - \alpha) \cdot Q_k(i, u) + \alpha \cdot c(i, u)$

So finally this means the use of leaving the final state is more expensive like staying within the goal state.

1.2 Q-Values during Episode

1.2.1 First run

Because the initial state has a $Q_{k=0} = 0$ due to no previous movement, we can calculate the next steps q-value as follows:
$Q_{k=1}(s_{1,1}, S) = (1 - \alpha) \cdot Q_{k=0} + \alpha(c(s_{1,1}, S) + \gamma \times Q_0(s_{2,1}, W))$
$Q_1(s_{1,1}, S) = 0 \cdot 0 + 1 \cdot (1 + 0.5 \cdot 0) = 1$
$Q_2(s_{2,1}, E) = 0 \cdot 0 + 1 \cdot (1 + 0.5 \cdot 0) = 1$
$Q_3(s_{2,2}, N) = 0 \cdot 0 + 1 \cdot (1 + 0.5 \cdot 0) = 1$
$Q_4(s_{2,2}, E) = 0 \cdot 0 + 1 \cdot (1 + 0.5 \cdot 0) = 1$
$Q_5(s_{2,3}, N) = 0 \cdot 0 + 1 \cdot 1 = 1$

1.2.2 Second run

Because the cost values didn't change $S_{1,1}$ still has a $Q = 0$ the improved Q-function after the initial episode is still the same.

Sources

[1] https://cs.stanford.edu/people/karpathy/reinforcejs/index.html