Q-Learning

Parameters used in Q-Learning

Parameters	Details
Episodes (Maximum Iterations) / Convergence Policy	Each episode consists of executing a path (Q-value updation) till the robot reaches the goal or falls in pit or reaches a maximum of 250 steps. We used the iterative approach of increasing the count (k) till 500. Also, the user can stop it with the stop button by visualizing the Q-values.
Exploration (ε)	Robot explores with the random action with probability P(ε) and goes to the desired action state with probability 1 - P(ε). In our Grid World setting, exploration value (ε) is set to a default value of 7 and a slide bar is provided to the user, to choose the exploration value ranging from 0 to 10 (0% to 100%).
Learning Rate (α)	Robot learns about the new state with a proportion of α times and memorizes the old values by a proportion of 1 – α. We choose the learning rate as a constant value of 0.80 in deterministic world. However, in stochastic world, it is a decreasing function ($\alpha = k^{-1}$) because the robot has learned enough and reached the point of convergence (less uncertainty).
Discount (γ)	Robot receives discount times the reward of next state. High value of Discount (1) is desired to get delayed reward. Our setting uses a discount value = 0.80.

Q-State: State achieved when the robot tooks the action a from state s, but hasn't reached the next state s' yet. It can either land in state T(s,a) (always in deterministic) or lands in some other state with a random probability (stochastic).

Transition Function: Transition function tells the robot the next state (s') after taking action (a) in state (s). In Deterministic World, Robot lands with 100% probability in the next state (s') after taking action (a) in state (s). However, in our Stochastic World, Robot lands in the desired state (s') with only 60% probability. Robot takes the other three actions which is not optimal with 10% probability and remains stationary with probability of 10%.

Deterministic World

$$Q(s,a) = R(s,a) + \gamma * max_{a'}Q(s',a')$$

$$Q(s,a) := (1 - \alpha) * Q(s,a) + \alpha * (R(s,a) + \gamma * max_{a'}Q(s',a'))$$

Stochastic World

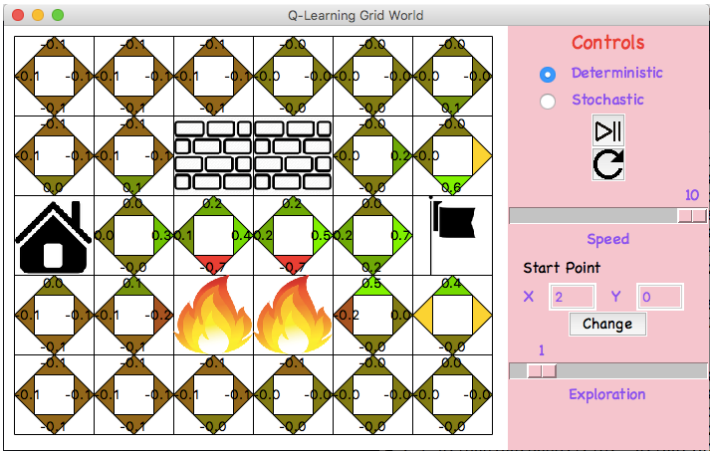
$$Q(s,a) = R(s,a) + \gamma * \sum_{s'} T(s,a,s') * max_{a'}Q(s',a')$$

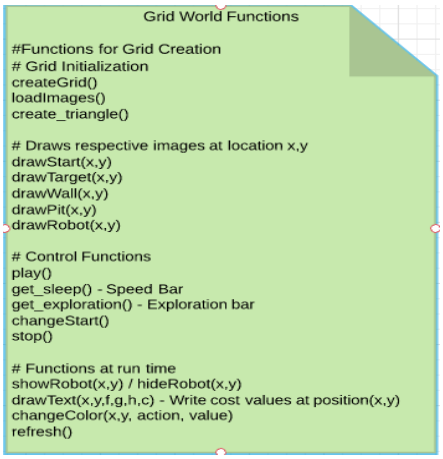
$$Q(s,a) := (1 - \alpha) * Q(s,a) + \alpha * (R(s,a) + \gamma * \sum_{s'} T(s,a,s') * max_{a'}Q(s',a'))$$

Reward Function R(s,a): Robot receives a Reward of going from one state (s) to another state (s') by taking an action (a). Robot receives the highest reward when it reaches the goal and receives the highest penalty when falls in pit. Also, every move of the robot counts, otherwise the robot will continuously explore the world ignoring the highest reward. This move cost is living reward (/walk reward) which is less in comparison to the highest reward. In our grid world setting, living reward is -0.05, goal reward = 1 and pit penalty = -1.

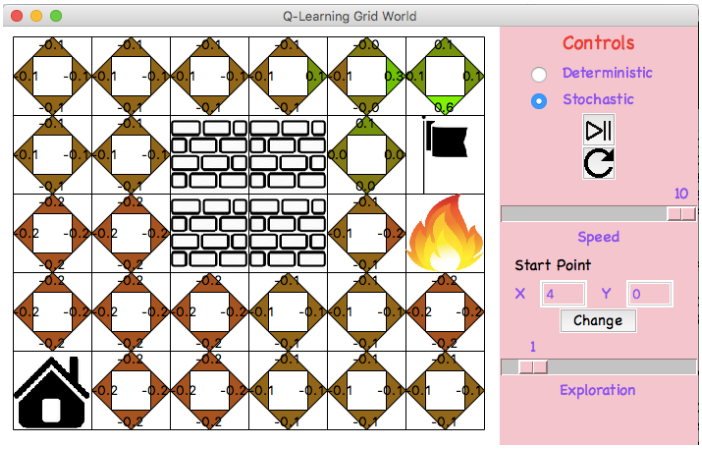
Exploration Policy: We generate a random value between 0 and 10 and if the value is less than exploration value, then the robot selects a random action with 25% probability of going in each direction (total of 4 directions). If the generated random value is greater than the explored value, then the robot takes the desired optimal action.

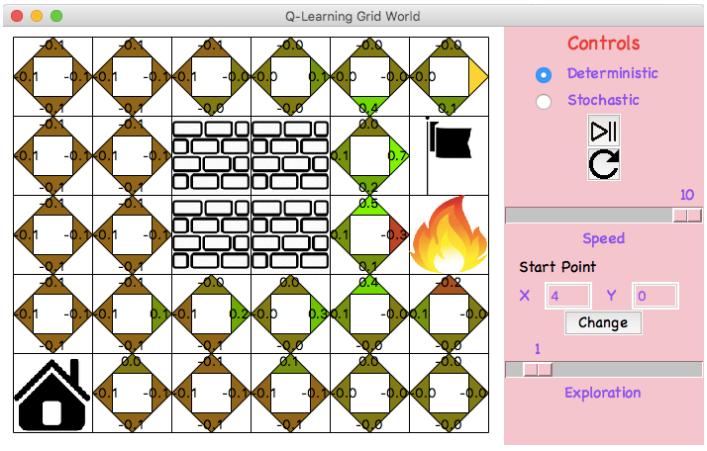
Heat Map: Colour changes from yellow to green for goal and from yellow to red for pit.





Example:





Note : Please find the Videos at:

1. A* in Grid World https://www.youtube.com/watch?v=xxZjoAqH3cU
2. Q-Learning https://www.youtube.com/watch?v=Nmx7gAUzw4M&feature=youtu.be