Reinforcement Learning Assignment

In this assignment you are to implement an agent that can walk down a sidewalk that contains litter to be picked up and obstacles to be avoided. To solve this you should use separate reinforcement modules, one for litter, one for obstacles, and one for staying on the sidewalk.
You also will need a forth module that guides the agent from the starting position and rewards the agent for making progress to the other end.

The sidewalk can be represented with a a rectangular path of discrete tiles. Actions can move the agent from the current tile to an adjacent tile.

The policy uses a reconciliation of the policies of the individual modules such as a reward-weighted sum or a soft max.

Your report should include graphics that illustrate the paths taken by the agent in the description of the simulation.

Details

As a baseline, however, consider a6x25 grid world. One square in the first column is the start position. All squares in the final column are terminating conditions. Each square is initialized with a 1/5 probability of being an obstacle. It's worthwhile to try different widths and obstacle densities.
The goal of the agent is to walk to the end of the sidewalk as fast as possible, while avoiding the obstacles. The agent has the ability to step on a square that has an obstacle on it, but it will receive a punishment (ie. a negative reward). The policy the agent learns should work on a variety of sidewalks, where the shape of the sidewalk is always the same, but the positions and number of obstacles varies.

Each module will be trained independently using Q-learning, using the epsilon-greedy strategy to pick the action at each time step. Given the current state, pick the best (ie. has highest q-value) action with probability epsilon (you get to choose this value, but 0.9 seems reasonable), and a randomly selected action (with uniform probability) with probability 1-epsilon. Training will consist of a sequence of episodes that ends when a convergence criterion is reached. Each episode is not a single action -- it is a sequence of actions that ends when some termination criterion (that you determine) is reached.

Here is how to combine the modules. Given the current state, pick the action that has the greatest weighted average q-value across the two modules. The weights you pick control the strength of each module in influencing the behavior of the agent. However, there is one problem with this strategy. The approach module's Q-values will increase with proximity to the goal, and will therefore unduly have more influence. You can come up with a normalization strategy to deal with this problem.

Once your modules are trained, you can use them in combination, by choosing an action that combines the recombinations of the different modules.