Assignment-2 Wumpus World

Group:

Gautham Nekkanti

Ramyasree Kola

We submit the wumpus world assignment aiming B grade (using reinforcement learning).

Reinforcement learning concerns with the agent performing actions in a given environment and maximizes their reward for that state action pair.

We used Q learning algorithm, which is a reinforcement learning algorithm that seeks to find the best action to take given the current state.We initialize Q table matrix which has scores for all possible actions in a given state with zeros. Each of these values will be maximum expected future reward that agent will get if it takes that action at that state.This is an iterative process, as we will be updating the q table at each iteration.

Code:

In GUI java file we give a new command action called “run simulation” where we initially try to load the q table data file if exists. And define an arbitrary count to be repeatedly run a “doAction” method on all the maps defined using the loaded q table.If the Q table file does not exist, we return an empty q table(which is the initial case and load it)

In the doAction() method, we first check for the default actions before performing optimal actions like grabbing gold / climbing pit. Then we create a state object and load(loadStateFromWorld) the current world instance into that state. In this loadStateFromWorld() we gather the current state percepts and then explore the neighbour tiles and second level of neighbours tiles and store their precepts as well.

After state loading, we lookup the Q table for actions possible for given loaded state; if no possible actions available in the Q table for that state we create an entry in the table.Then we choose the best action from these possible actions.Choosing the highest scored possible actions.

We use a technique called ‘exploration-exploitation tradeoff’ Using this we guarantee that the agent does not always go for the small reward options leaving any chance of achieving far-off greater rewards, So we trade off between continuous exploitation(goin for smaller rewards) and rather exploring other paths. So we use an epsilon values to choose between these and select an optimal action and perform it. After doing that, we then load a new state at the instance after action performed.And calculate the reward for the action performed.This is calculated as per the assignment requirements.Finally we save the loaded q table into the file. This is done for simulation.

While running the agent , we use the loaded q table to choose the optimal actions.

//state percepts are stored

//n and nn