PROJECT 4 – Q-TABLE BASED REINFORCEMENT LEARNING

# qTableConfig Function Class

* Contains the values of PITREQARD, WUMPUSREWARD, GOLDREWARD, BUMPREWARD.
* Other functions (TrainAgent) call upon this class when they require any of these default values.

# AgentTraining Function Class

* This is the function used for training the Agent before the actually execution of the Agent in Wumpus Environment.
* The agent keeps on taking random actions and checks the utility for each action taken.
* These utility values are propagated back to the previous state.
* When learning is finished, the agent will choose the action based on best value at every state.
* These state-action utility and state utility values associated with every state and every possible action at that state are called QValues.
* Our AgentTraining Class stores these values into a Q-Table.
* Thus, we now have a q-table to use for our actually agent execution run.

# qTable Function Class

* Initializes a qTable for each new instance with value set to 0 and passes this ‘empty’ set to the calling functions (TrainAgent).
* It has 3 other functions:
  + getBestActionMaxQValueAction: Finds the next safe/best place for the agent by scanning and searching through the qTable to get the best max value of the table,
  + getMaxQTableValue: Finds the qTable element with the best/max value to be used in the next agent movement.
  + Display Table: Shows the final qTable we used for our Agent’s movement. (This function has only ever been used for debugging and verification. Currently, it does not display anything).

**State update based on incoming percept, and action performed (Function Classes Agent, AgentFunction and Action) is same as in case of model based agent.**

# Updating model based on agent’s action - Action ( )

• World model will be updated every time based on the next action taken.

1. Next Action - GO\_FORWARD

• Update agent’s location variable to the location which is one step ahead in the direction it is facing, only if it is a valid location on the grid.

2. Next Action - TURN\_LEFT

• Update agent’s direction to the new direction obtained by turning left from the current direction. This is done by using a static dictionary *turn<Direction,Direction>*

3. Next Action - TURN\_RIGHT

• Update agent’s direction to the new direction obtained by turning right from the current direction. This is done by using a static dictionary *turn.*

4. Next Action - SHOOT

• Get the direction the agent is facing. Update the Presence of Wumpus in all valid locations straight in that direction to *False.* Since, an arrow has been shot, if the wumpus was in any of the locations straight ahead in that direction, it will be killed with 100% chance.

5. Next Action - NO\_OP

• No modifications to the world model.

**Output -** For 10000 trials and 1000 training iterations, the qTable Reinforcement Learning agent takes approximately 10 mins to run.

Total Score: 311689

Average Score: 31.1689

Total Score: 384044

Average Score: 38.4044