دانشکده مهندسی کامپیوتر
گروه هوش مصنوعی

استاد درس: حسین کارشناس
دستیار درس: پوریا صامتی

MDP Environment

Actions			
Up: 0	Down: 1	Left: 2	Right: 3

Tile Type in Grid				
Queen: Q	Pig: P	Rock: R	Tile: T	Goal: G

Function	Information	Return object
env.reset()	This function resets the environment, starting a new episode with the initial state of the environment.	(0, 0)
env.step(action)	This function takes an action, moving the agent to the next state based on specified action probabilities. It returns the next state (with the agent's position as a tuple, e.g., (1, 2)), the probability of the selected action, the reward associated with the action, and a boolean is_terminated indicating if the episode has ended.	next_state, probability, reward, is_terminated
env.reward_function()	You should implement this function to return an 8x8 matrix, where each element represents the reward for transition between tiles.	8x8 Matrix
env.render(screen)	This function displays the game environment and can be called within the game loop.	-

Attributes	Information
env.transition_table	The **transition_table** is a comprehensive dictionary that defines the possible outcomes for each action the agent can take from every state in the environment. It's essential for understanding how the agent interacts with its environment and calculates the consequences of its actions. **Structure**: • **Keys (States)**: Each key in the **transition_table** represents a state (row, col). For example, (2, 3) represents the position of the agent in the grid. • **Values (Actions)**: For each state, there's another dictionary where each key is an action (0 for Up, 1 for Down, 2 for Left, 3 for Right). **Action Outcomes**: • For each action at a given state, we have a **list of possible outcomes**. Each outcome in this list is represented by a tuple, containing: ○ **Probability**: The likelihood of this outcome if the agent attempts this action. ○ **Next State**: The resulting state after the action, given as coordinates (new_row, new_col). ○ **Reward**: The reward the agent receives for reaching this state after the action. We have an example -> The entry in **transition_table[(2, 2)][0]** might look like this: `[ (0.70, (1, 2), -4), (0.15, (2, 1), -3), (0.15, (2, 3), -3) ]` This means if the agent tries to move Up from (2, 2), it has a: • **70% chance** of reaching (1, 2) with a **reward of-4**. • **15% chance** of going Left to (2, 1) with a **reward of-3**. • **15% chance** of going Right to (2, 3) with a **reward of-3**. Pay attention to the reward values in this **transition_table**. These rewards are calculated by the **env.reward_function**(you should implement this before).

env.grid	The output is a grid of size 8x8, where each cell represents a specific tile type. This grid can be used as a map of the environment. Each time you call **env.reset()**, the environment is restructured according to this map. It is a matrix like this: `[['T', 'T', 'R', 'T', 'P', 'T', 'T', 'T'], ['T', 'Q', 'T', 'T', 'T', 'R', 'T', 'T'], ['T', 'T', 'T', 'T', 'P', 'T', 'Q', 'T'], ['P', 'T', 'T', 'R', 'T', 'T', 'T', 'T'], ['T', 'T', 'Q', 'T', 'T', 'T', 'P', 'T'], ['R', 'T', 'T', 'T', 'T', 'T', 'T', 'T'], ['T', 'P', 'T', 'R', 'T', 'T', 'Q', 'T'], ['T', 'T', 'T', 'T', 'T', 'T', 'T', 'G']]`
env.reward_map	This attribute stores the result of **env.reward_function()**. It is an **8x8 matrix** where each element represents the reward associated with a specific tile in the environment.