Quiz #2 - Results

Attempt 1 of 2

Written Sep 22, 2025 2:34 PM - Sep 22, 2025 2:47 PM

Attempt Score **1.2 / 1.5**

Overall Grade (Highest Attempt) **1.35 / 1.5**

Question 1 0.15 / 0.15 points

In the design of Intelligent Agents, these are hardware components

- ○ Sensors and Actions
- ○ Percepts and Actuators
- ✓ ◉ Sensors and Actuators
- ○ Percepts and Actions

Question 2 0.15 / 0.15 points

What is the minimum requirement for any device to be considered an intelligent agent?

- ✔ ◉ The ability to carry out activities which need the agent to pick non-trivial choices.
- ○ A humanoid body.
- ○ Completing tasks that can only be done well by thinking carefully.
- ○ Being at least as smart as a house pet.

Question 3 0.15 / 0.15 points

What is the role of perception when it comes to intelligent agents?

- ○ It plays a crucial role in intelligent agents, as it serves as the mechanism by which an agent gathers information about its environment. This information alone can help the agent pick actions that will maximize its performance metric.
- ○ The agent collects raw data from sensors, such as cameras, microphones, or other devices that mimic human senses. These sensors provide essential inputs about the environment, converting any partially observable environment into a fully observable one.
- ○ Intelligent agents rely on perception to make informed decisions in real-time. Perception allows the agent to adjust the environment.
- ✔ ◉ An intelligent agent will rely on sensors to provide information about its environment. The information can then help the agent build an internal representation of the environment, making it very useful in partially observable environments.

Question 4 0.15 / 0.15 points

Why do we say that rationality always depends on the agent's prior knowledge of the environment?

- ○ In many cases, an agent does not have complete knowledge of the

environment, so it must make decisions based on its prior information. This information is hard-coded and we, the designers, need to make sure that we include everything that the agent needs to know.

- ○ Because an agent can only make decisions based on what it knows or believes about its surroundings. Incorrect knowledge cannot be fixed and it will always lead to bad decisions.
- ✔ ◉ An agent that does not know the rules of the environment cannot make predictions of the outcomes of its actions and therefore it cannot make proper plans. Regardless, the agent still needs to act, and bad choices might still be considered rational if they were the most promising given the prior knowledge.
- ○ What constitutes a "rational" action can vary depending on what the agent knows about the environment. Not knowing the potential outcomes of a given action can deliver the agent from getting a lower performance score if the outcome is bad.

Question 5 0 / 0.15 points

In the implementation of Search algorithm, the following classes and/or functions are generic and will not change with the application:

- ✖ ◉ Problem, Search Algorithm
- ○ Search Tree Node, Problem
- ○ State, Search Tree Node
- ➡ ○ Search Tree Node, Search Algorithms

Question 6 0.15 / 0.15 points

We said that task environments must be deterministic, or we cannot use search algorithms. Why is this the case?

- ○ A deterministic environment can be easily simulated by the agent using deterministic time steps. This allows the agent to create plans that only consider a simple sequence of actions to perform. The actual time required to perform these actions might be easily absorbed as part of the action cost.
- ○ A deterministic environment means that there is only one possible result for every action. Thanks to this, the agent can make complex plans trusting the results of every action even when the rules of the environment are unknown.
- ○ A deterministic environment means that the agent is familiar with the rules of the environment. This is important because such rules help the agent to predict how particular actions might have specific effects on the environment. Without this information, the agent cannot predict the resulting states of different sequences of actions.
- ✓◉ A deterministic environment means that the agent can trust that performing the same action over the same state will always lead to the same outcome. Without this guarantee, the agent cannot predict the resulting states of different sequences of actions, as these might be different every time.

Question 7 0.15 / 0.15 points

An intelligent agent has used A* to create a plan to solve a particular problem. The agent can now blindly follow this plan, and the problem will be solved. True or false?

- ✓◉ It depends on the problem actually. If this problem meets certain conditions, the agent can safely ignore its sensory inputs. For many

applications, it cannot (and should not be using A* in the first place).

- False. Agents should always pay attention to their inputs in case that something unexpected happens.
- False. Agents need to use rules to react to the uncertainty of the environment.
- True. The plan considers all potential outcomes of all actions performed of the agent and regardless of the problem, the agent can safely follow this plan.

Question 8 0 / 0.15 points

We said that task environments must be known, or we cannot use search algorithms. Why is this the case?

- ✖ ◉ A known environment means that the agent can trust that performing the same action over the same state will always lead to the same outcome. Without this guarantee, the agent cannot predict the resulting states of different sequences of actions, as these might be different every time.
- ➡ ○ A known environment means that the agent is familiar with the rules of the environment. This is important because such rules help the agent to predict how particular actions might have specific effects on the environment. Without this information, the agent cannot predict the resulting states of different sequences of actions.
- ○ A known environment can be easily simulated by the agent using known time steps. This allows the agent to create plans that only consider a simple sequence of actions to perform. The actual time required to perform these actions might be easily absorbed as part of the action cost.
- ○ A known environment means that the agent knows the result of

every action. This allows the agent to deal with complex real-time environments.

Question 9 0.15 / 0.15 points

An agent has to solve a Sudoku by adding values one by one to the 9x9 grid. As usual, there will be some pre-set numbers in the grid, and the agent must complete the grid while respecting all the constraints of this game. A given state can be determined to be a goal state for this problem using:

- ○ A database of known sudoku solutions.
- ○ A function that will compare the solution provided by the agent to a preset solution from which the initial sudoku grid was generated. It will return false if these two are not identical.
- ✓ ◉ A function that will take the state as input and will return true only if all cells in the grid have a valid number, and every row, column, and sub-region of the grid has the numbers from 1 to 9 without repetitions. The function should return false if any of these conditions is not met.
- ○ A function that returns true if the grid is complete.

Question 10 0.15 / 0.15 points

In the maze navigation problem, an agent is trying to exit a maze represented by a regular grid. To navigate the maze, the agent can only move forward or backwards (1 cell each). The agent is always facing one out of four directions: North, South, East, West. The agent can rotate -90 or +90 degrees to change its current direction. Each rotation in place consumes 25% more battery than moving forward or backwards. There are thin but strong walls in this maze, which the agent cannot simply pass through. The agent has to move from its current position at (S_row, S_col)

to a target location (T_row, T_col). The Action Cost function for this problem can be defined as follows:

- ○ Cost(action) = {125% if action in ["Move Forward", "Move Backward"] else 100%}
- ○ Cost(action) = 1
- ○ Cost(action) = 125%
- ✔ ◉ Cost(action) = {125% if action in ["Rotate -90", "Rotate +90"] else 100%}

Done