DQ1 (L1)

Due 15 Jan at 23:59 **Points** 25 **Questions** 11
Available after 9 Jan at 12:00 **Time limit** None
Allowed attempts Unlimited

Instructions

- This quiz is NOT GRADED. However, it is HIGHLY RECOMMENDED that you use these questions to complement your review of the lecture content.
- The questions are based on content from the Lecture 1 and from Chapter 1 and 2 of the AIMA (4th Ed.) textbook.

Take the quiz again

Attempt history

	Attempt	Time	Score
KEPT	**Attempt 4**	2 minutes	25 out of 25
LATEST	**Attempt 4**	2 minutes	25 out of 25
	Attempt 3	1,689 minutes	0 out of 25
	Attempt 2	8 minutes	25 out of 25
	Attempt 1	18 minutes	17.67 out of 25

Submitted 15 Jan at 23:27

Question 1 — 2 / 2 pts

Which of the following systems involve artificial intelligence (assuming that reflex agents are excluded)?

Choose all relevant options.

- [ ] Simple inventory/stock identification/tracking systems based on bar codes or RFID tags (e.g., those you typically see in most supermarkets in Singapore today). (Note that the above system includes the restocking of shelves, but not ordering new stock from any distributors.)

Correct! - [x] Hands-free voice-activated mobile phone interfaces.

Correct! - [x] Online search engines.

Correct! - [x] Game mechanics in popular MMORPGs.

- **Inventory systems** = No. Essentially map physical items with tags and identifiers, typically there is no need to for a dynamic system on top of this.
- **Voice-activated interfaces** and **search engines** = Yes. Such systems are typically made more accurate by being customised to a given user (since each user may have a different accent, use different languages, or possess a different set of knowledge and understanding of the concepts included in that knowledge), which implies that they require more dynamic solutions. For example, an internet search for the keyword ***model*** made by a fashion designer versus a computer scientist (assuming that they are both focused on work) should yield very different results.
- **MMOPRGs** = Yes. In terms of simple functionality such as path finding. However, in terms of agents that play certain roles, these are mainly reflex since most MMORPGs are focused on more story-line and rely on static rules - i.e., agents in such environments are typically reflex-based at best, even though some function contains traces of AI.

Question 2

2 / 2 pts

Why is it so difficult to develop **general** Artificial Intelligence?

Select the best option.

It is too expensive to develop an agent that knows how to solve everything.

Correct!

It is difficult to define and specify all human-relatable tasks and the solutions that can be solved by a general AI.

Companies must seek permission from relevant government authorities to establish a general Artificial Intelligence agent.

A general AI must define a specific problem, which is challenging.

All of them are actually valid, but the second point is the root cause of the difficulty in establishing general AI.

- Option 1: While this is true, we want the best option, and cost is secondary if even enumerating ***everything*** is almost virtually impossible
- Option 3: Same rationale as above, difficult to quantify ***general***
- Option 4: Even if you can establish problem, we need a ***general solution***

Question 3

2.5 / 2.5 pts

Suppose we designed and implemented a system, that when given a new problem, would be able to:

1. categorise the kind of problem at hand, and
2. utilise a separate and independent narrow AI solution to solve that problem.

Such a system be deemed to be a strong (i.e., general) AI.

True

Correct!

False

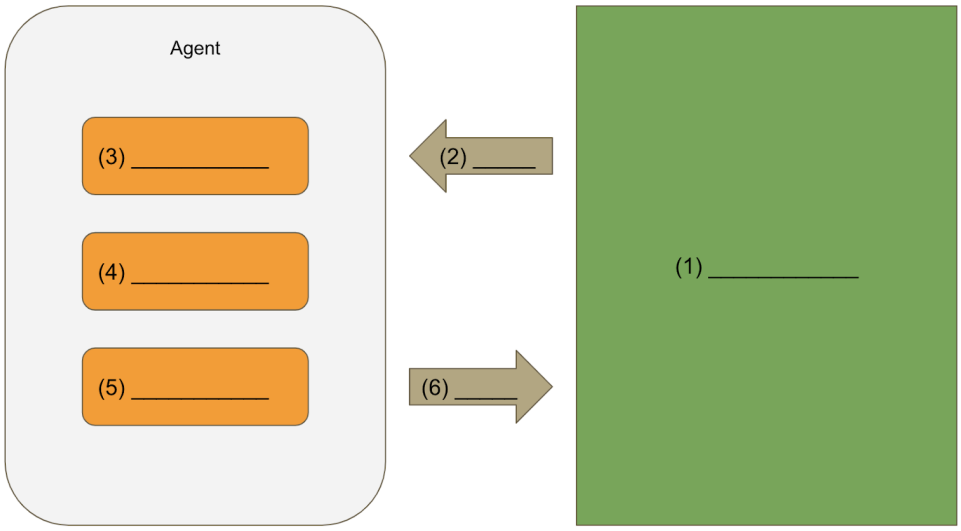
Such a system could indeed be argued to be less narrow AI as compared to another system that focused on one very specific problem. For example, a system that just played Chess, versus one that included two independent solutions, one to play Chess and another to play Go. However, the problems with such systems is that each solution is separate, and as such, would be doomed to fail when approaching true generality (a very large number of problems) since there would be far too many narrow AI solutions that would have to be constructed.

The main idea behind strong AI is generalisation. That is, the ability to generalise between problems such that solutions are shared across problems. For example, AlphaGo versus AlphaZero. Refer to this article: **https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go (https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go)**

Question 4

1.5 / 1.5 pts

Fill in the corresponding blanks with the correct labels for the **structure of an Intelligent Agent**.



(1) Problem Environme

(2) Percepts

(3) Sensors

(4) Function

(5) Actuators

(6) Actions

Do not leave any trailing blanks or characters after your answer.

Answer 1:

Correct! Problem Environment

rrect answer Environment

rrect answer environment

rrect answer problem environment

Answer 2:

Correct! Percepts

rrect answer percepts

rrect answer Percept

rrect answer percept

Answer 3:

Correct! Sensors

rrect answer sensors

rrect answer Sensor

rrect answer sensor

Answer 4:

Correct! Function

rrect answer function

rrect answer Functions

rrect answer functions

rrect answer Agent Function

rrect answer agent function

Answer 5:

Correct! Actuators

rrect answer actuators

rrect answer Actuator

rrect answer actuator

Answer 6:

Correct! Actions

rrect answer actions

rrect answer Action

rrect answer action

As definied by the answers, which are based on Slide 21.

Question 5

2 / 2 pts

The **agent function** is an abstract mathematical function that maps every possible percept sequence to an action .

Answer 1:

Correct! Percept Sequence

orrect answer percept sequence

orrect answer Percept Sequences

orrect answer percept sequences

orrect answer Sequence of Percepts

orrect answer sequence of percepts

orrect answer Percept History

orrect answer percept history

Answer 2:

Correct! Action

orrect answer action

Since the agent here is general, we may assume that it is to be applied on a problem environment that is either sequential or episodic. Such an agent thus requires a percept ***sequence*** to be mapped to each action.

Question 6

2 / 2 pts

An agent that **senses only partial information** cannot be perfectly rational.

True

Correct! False

This is **False** since perfect rationality refers to the ability to make good decisions ***given*** the ***sensor information received***.

Question 7

2 / 2 pts

A **perfectly rational** poker-playing agent **never loses**.

- True
- Correct! False

False. Pit two of the same perfectly rational poker-playing agent against each other. One of them will lose.

Question 8

2 / 2 pts

Which of the following games are **Episodic**?

Assume that you play with anonymous, arbitrary opponents.

- Chess
- Correct! Roulette
- Correct! Scissor-Paper-Stone (also known as Rock-Paper-Scissors)
- Tic-Tac-Toe

Recall the definitions of episodic and sequential.

- Episodic → actions only impact the current state and not those beyond
- Sequential → an action may affect all future decisions made by the agent

For games like Chess and Tic-Tac-Toe, since each move affects all future states, they are sequential. However, with Roulette and Scissors-Paper-Stone are episodic since each action made only determines the state of the round in question and has no bearing on future rounds.

Question 9

2.5 / 2.5 pts

Select the options that describe the **environment characteristics** of a game of **Tic-Tac-Toe**.

- [ ] Tic-Tac-Toe is partially observable
- [x] Tic-Tac-Toe is deterministic — Correct!
- [x] Tic-Tac-Toe is sequential — Correct!
- [x] Tic-Tac-Toe is discrete — Correct!
- [ ] Tic-Tac-Toe is single agent

Tic-Tac-Toe:

- Fully-observable - i.e., there is no hidden information
- Deterministic - i.e., there is no randomness governing states, transitions, etc.
- Sequential - i.e., each move affects all future states
- Discrete - i.e., each state is fully described with discrete values
- Multi-agent - i.e., there is a competing agent

Question 10

2.5 / 2.5 pts

Select the options that describe the **environment characteristics** of a game of **Poker**.

Correct! ☑ Poker is partially observable

☐ Poker is deterministic

Correct! ☑ Poker is sequential

Correct! ☑ Poker is discrete

☐ Poker is single-agent

Poker:

- Partially-observable - i.e., deck and opponent's cards are hidden
- Stochastic - i.e., we cannot determine which actions (fold, call, raise, etc.) the opponent will choose to make
- Sequential - i.e., each move affects all future states
- Discrete - i.e., each state is fully described with discrete values
- Multi-agent - i.e., has one or more competing agents

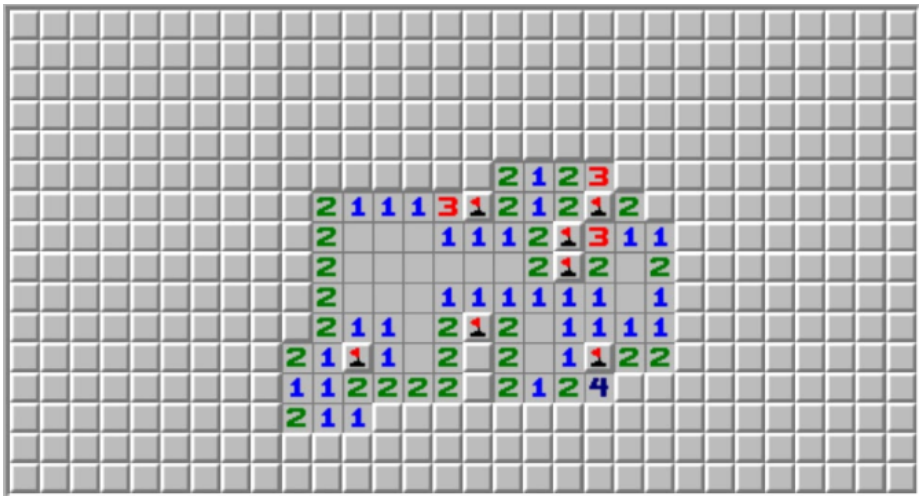
Question 11

4 / 4 pts

Match each problem environment to the most appropriate agent type.

Some assumptions:

- **General** (i.e., for all four games)
  - Assume only one game to be played
- **Tic-Tac-Toe** (also called noughts and crosses or Xs and Os)
  - Assume game is played against a human
- **Minesweeper**
  - Assume game is played on one specific map — for example:



- **Maze Navigator**
  - Assume any maze configuration may be given
  - Assume there is always at least one exit
  - Assume that the start is always connected to at least one exit
  - Assume the maze remains unchanged while being solved
- **Chess**
  - Assume game is played against a human

Correct! **Tic-Tac-Toe** Reflex Agent

Correct! **Minesweeper** Model-based Reflex Ag

Correct! **Maze Navigator**

Goal-based Agent

Correct!

Chess

Utility-based Learning A

- **Tic-Tac-Toe:** This game may be played optimally via a fixed set of rules.

  Refer to **https://en.wikipedia.org/wiki/Tic-tac-toe#Strategy (https://en.wikipedia.org/wiki/Tic-tac-toe#Strategy)** .

- **Minesweeper:** This game requires one to keep an internalised model (e.g., marking definite and possible mines) and then follow a fixed set of rules to reveal more and more of the board until the whole board is uncovered.

  We will study how to deal with incomplete knowledge when we cover logical agents, which can be applied to Minesweeper.

- **Maze Navigator**: This game requires the use of a path-finding algorithm to determine a path from the start to the exit. We require a definition of the goal (i.e., the exit), and will thus conduct a search for a path from the starting position to that goal.

  We will look at these sorts of problems when we cover uninformed search in the next lecture.

- **Chess**: To play chess, we need an agent similar to that of AlphaGo, these are utility-based, learning agents.

  While we will look at utility-based agents when we cover adversarial search, we will not cover learning agents in CS3243. You can require reinforcement learning agents as independent learning if you wish.