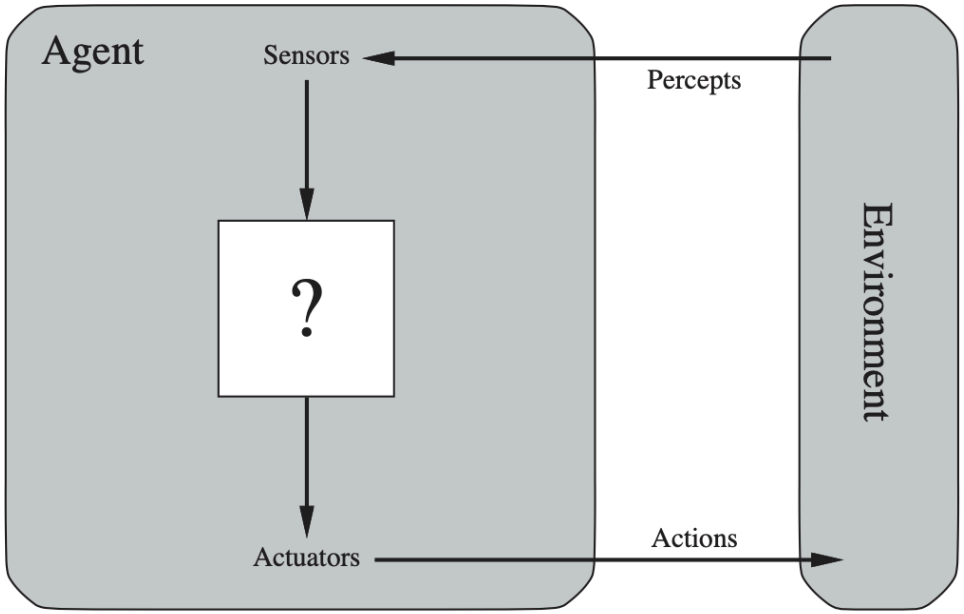
# CS143: Artificial Intelligence

## Intelligent Agents and Environments




Drake UNIVERSITY

# Announcement and Agenda

- Week 1 small assignment#1 on Blackboard.
- Discussion of intelligent agents, following the presentation from the second chapter of *Artificial Intelligence: A Modern Approach* by Russell and Norvig.
- By the end of today's class, you should have a basic familiarity with:
  - How an agent interacts with an environment.
  - The concept of rationality.

# Recap

- Course introduction and syllabus.
- We briefly surveyed key developments in the history of AI.
- We talked about different approaches to defining AI and identified our approach: AI as the pursuit of rational action.

# Recap: What is AI?

| Think like humans | Think rationally |
| --- | --- |
| Act like humans | Act rationally |

- The goal of the discipline AI is: to build machines that ~~perform tasks in a way similar to an intelligent human~~ make the "best" decisions given current knowledge and resources

# Recap: Open Philosophical Question: Are Machines Intelligent?

- **A1:** Yes, if intelligence is narrowly defined as "information processing"
  - Many tasks initially assumed to require intelligence can be automated.
  - Each success of AI seems to push further the limits of what we consider "intelligence"

- **A2:** Maybe yes, maybe not, if intelligence cannot be separated from consciousness
  - Is the machine *experiencing* thought?
  - Strong vs. Weak AI

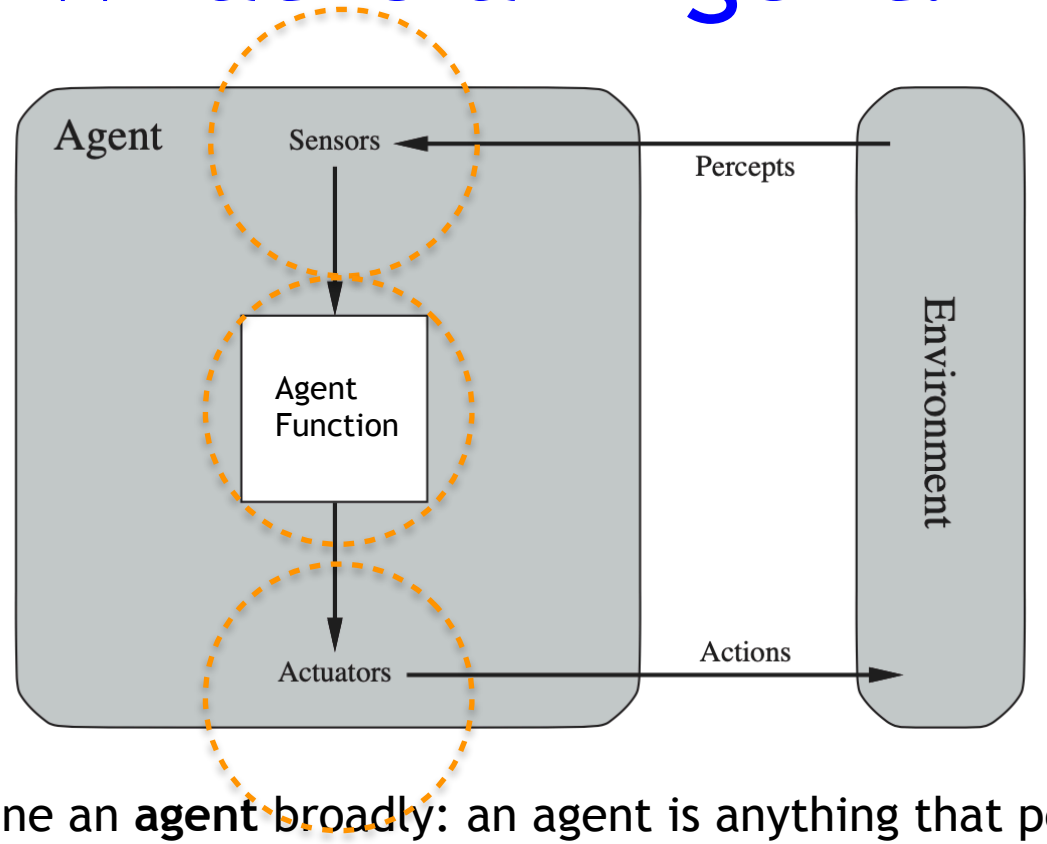
# Intelligent agents: what is rationality?

# What is a rational agent?

- An **agent** is an entity that perceives and acts.

- Our course is about designing a **rational agent**; that is, we seek to design an agent that can make the **best decision** under a given set of resources/ circumstances.
    - Let's first a define agent
    - Then define a rational agent

# What is an Agent?




- We will define an **agent** broadly: an agent is anything that perceives its environment via **sensors** and acts upon the environment with **actuators**.
- **Sensors** receive percepts (inputs), while **actuators** yield actions (outputs)
- This characterization suggests that our agent is a robot (as it uses the terms sensors and actuators), but it is much broader than that.

# Example: Agents

- Example 1: Humans
  - Sensors:
  - Actuators:

# Example: Agents

- Example 1: Humans
  - Sensors: Eyes, ears, other sensory organs
  - Actuators: Hands, legs, vocal chords, etc.

# Example: Agents

- Example 1: Robots
  - Sensors:
  - Actuators:

# Example: Agents

- Example 2: Robots
  - Sensors: Camera, distances sensors, microphones
  - Actuators: mechanical arms, motors, lights

# Example: Agents

- Example 1: Software agents
  - Sensors (sensory inputs):
  - Actuators:

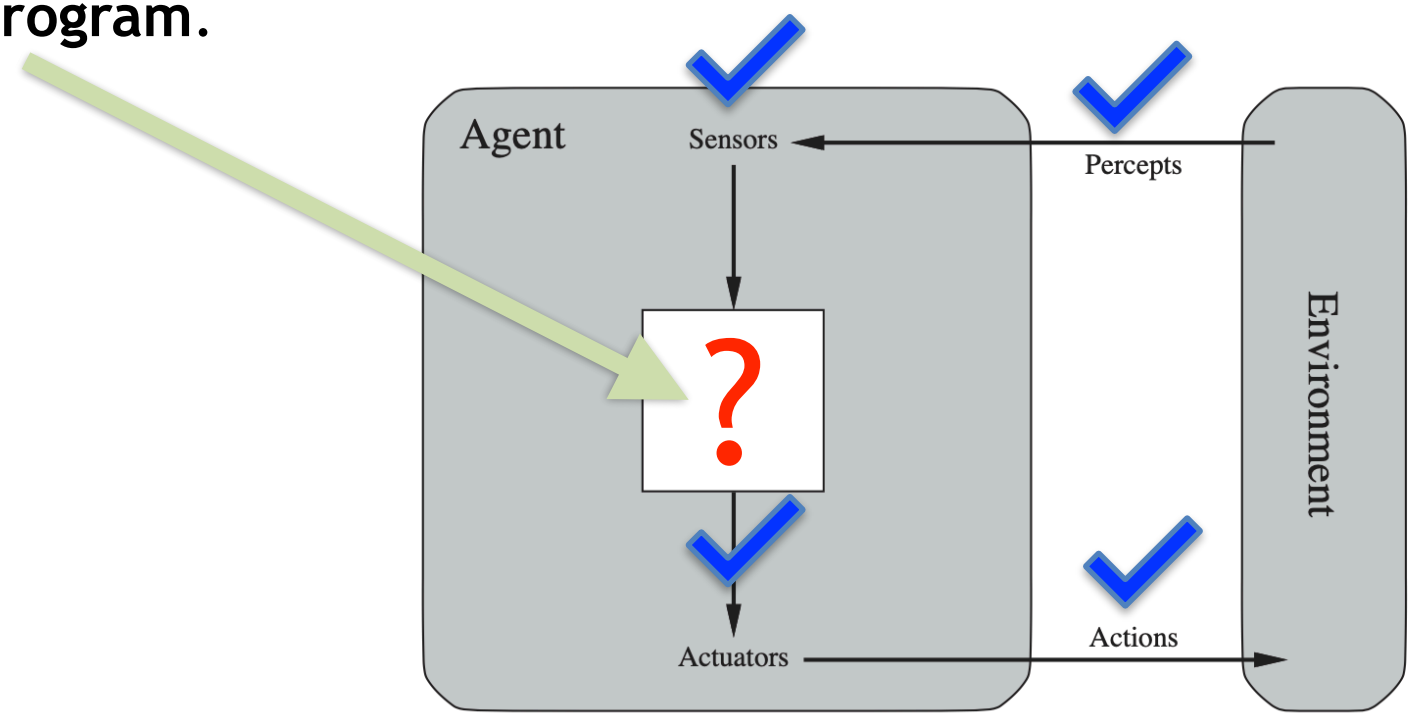
# Example: Agents

- Example 3: Software agents
  - Sensors: Receiving keystrokes, file contents, network packets
  - Actuators: Displaying on the screen, writing files, sending network packets

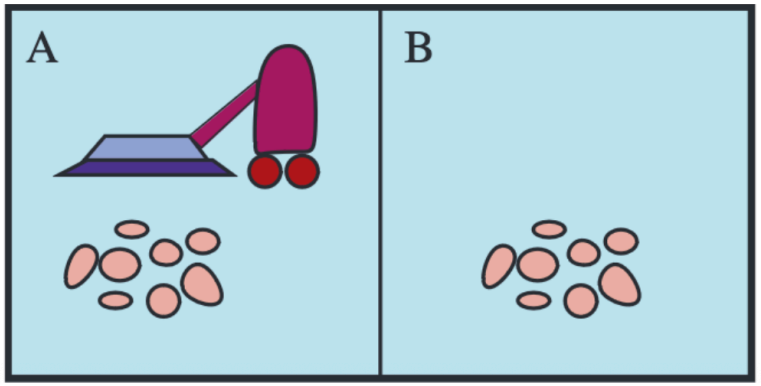
# Agent: Agent function

- An agent's behavior is described by an **agent function**, which is a function that maps every possible percept sequence to an action.

- we can *internally* characterize an agent by means of a program that implements the **agent function**, which we call an **agent program**.



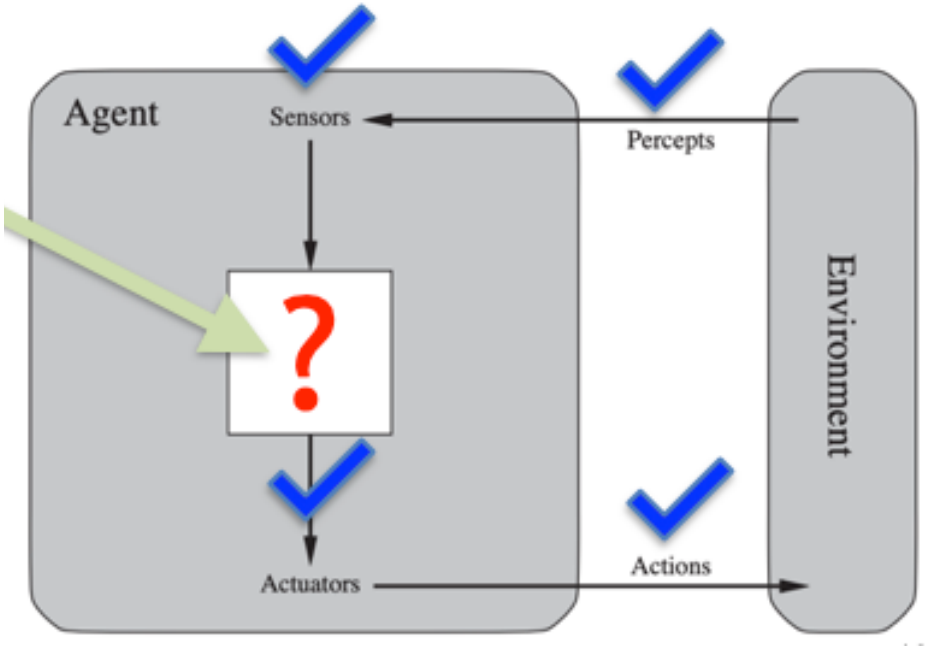
# Agent function: vacuum-cleaner agent example




- **Agent:** Vacuum agent
- **Environment:** it can occupy squares *A* and *B*, which can either be dirty or clean
- **Percepts:** The vacuum agent can perceive which square it is in (location) and whether it is clean or dirty (contents)
  - Eg, [A, Dirty]
- **Actions:** The vacuum agent can move to the left or the right, it can suck up dirt, or it can do nothing.
  - Eg, Left, Right, Suck, and NoOp

# Example: Agent function in terms of a program

- "If the current square is dirty, then suck; otherwise, move to the other square."
- Here's an agent program that implements this agent function.




**function** REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

**if** *status* = *Dirty* **then return** *Suck*
**else if** *location* = *A* **then return** *Right*
**else if** *location* = *B* **then return** *Left*

# Agent function in terms of a table of values

- Here is a partial table of values given by our choice of agent function:

| Percept sequence | Action |
|---|---|
| $[A, Clean]$ | $Right$ |
| $[A, Dirty]$ | $Suck$ |
| $[B, Clean]$ | $Left$ |
| $[B, Dirty]$ | $Suck$ |
| $[A, Clean], [A, Clean]$ | $Right$ |
| $[A, Clean], [A, Dirty]$ | $Suck$ |
| ⋮ | ⋮ |
| $[A, Clean], [A, Clean], [A, Clean]$ | $Right$ |
| $[A, Clean], [A, Clean], [A, Dirty]$ | $Suck$ |
| ⋮ | ⋮ |

- Are some ways better than others ‘Tabular’ vs ‘Function’

# What is a rational agent?

- An **agent** is an entity that perceives and acts.

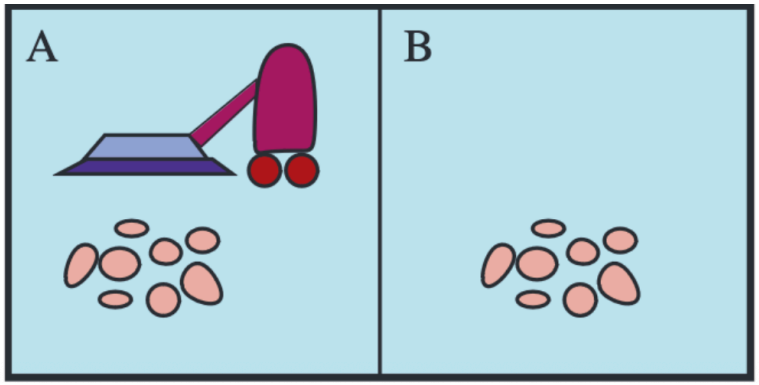
- Our course is about designing a **rational agent**; that is, we seek to design an agent that can make the **best decision** under a given set of resources/ circumstances.
  - Let's first a define agent
  - Then define a rational agent

# How do we measure rationality?

- Recall that we defined **a rational agent** to be an agent that "takes the best decision" or "does the right thing"
- But what exactly is the "best decision"/"right thing"?
- We answer this by considering the consequences of the agent's behavior

# Performance measure: vacuum-cleaner agent example

- Fixed **performance measure** evaluates the environment's sequence of states. If the sequence is desirable, the agent has performed well:
  - **Measure#1:** Measure performance by the amount of dirt cleaned in an eight hour shift.
    - One point per square cleaned up in time T?
  - **Measure#2:** Measure performance by rewarding the agent for having a clean floor
    - One point per clean square per time step, minus one per move?



# Insight for Performance Measure

- It is better to design performance measures according to a desired outcome for the environment as opposed to how one thinks the agent should behave.

- "Performance measure#1" was based on how much cleaning the agent does, not on how clean the environment is.

- "Performance measure#2" was based on maintaining a clean environment (and avoided the above problem).

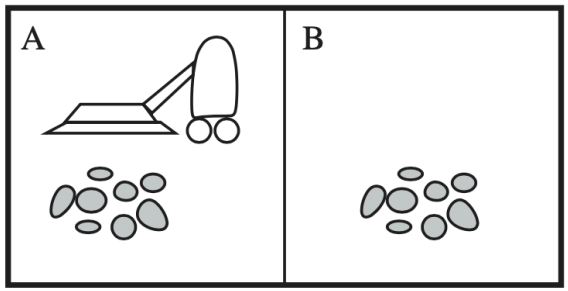
# Precise definition of rationality

- We can now define a rational agent as follows:
  - A rational agent is one that selects an action that maximizes its expected (or average) performance measure given the percept sequence to date.

# Vacuum-cleaner's rationality

- Is the vacuum agent defined by the following agent function rational?

```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```




  - "Yes": If we simply want a clean environment
  - "No": If we want a clean environment while minimizing the number of steps.

# Task environments for a rational agent

# What is task environment?

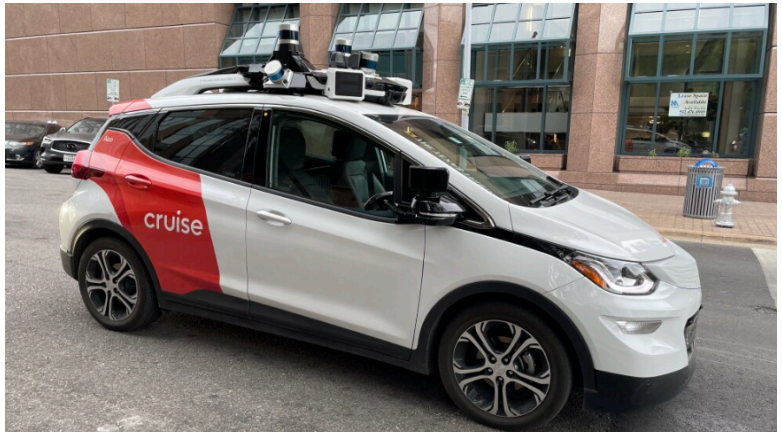
- To design a rational agent, we must specify task environments, which can be thought of as the *problems* to which the rational agents are the *solutions*

- We define a task environment by specifying four factors:
  - **P**erformance Measure
  - **E**nvironment
  - **A**ctuators
  - **S**ensors

- We call this the PEAS description of a task environment

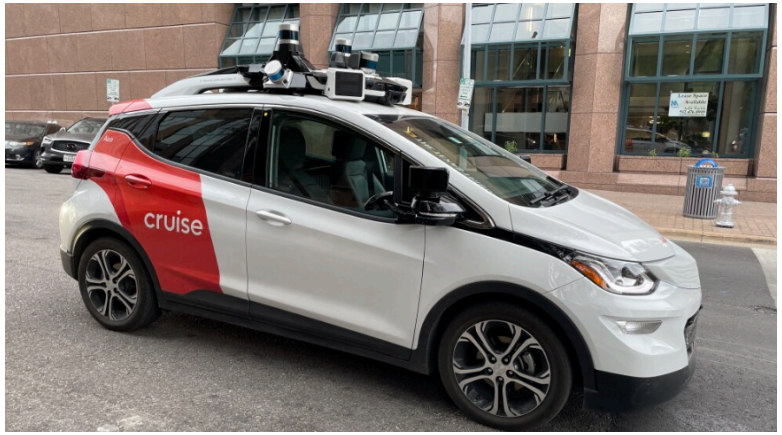
# PEAS example: automated taxi driver

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | | | | |



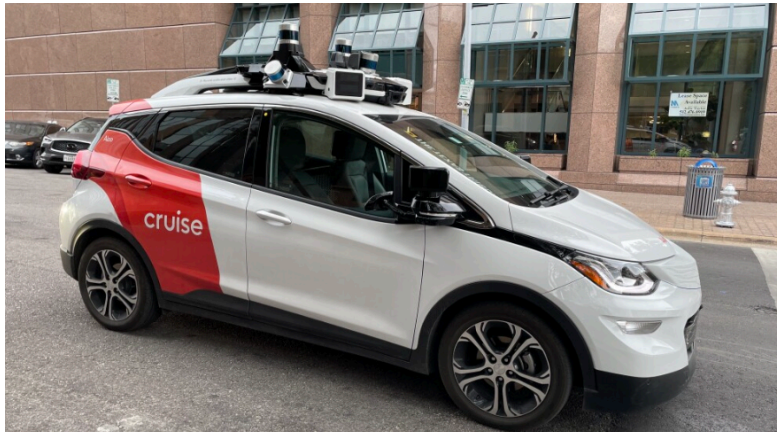
# PEAS example: automated taxi driver

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | Safe, fast, legal, comfortable trip, maximize profits | | | |



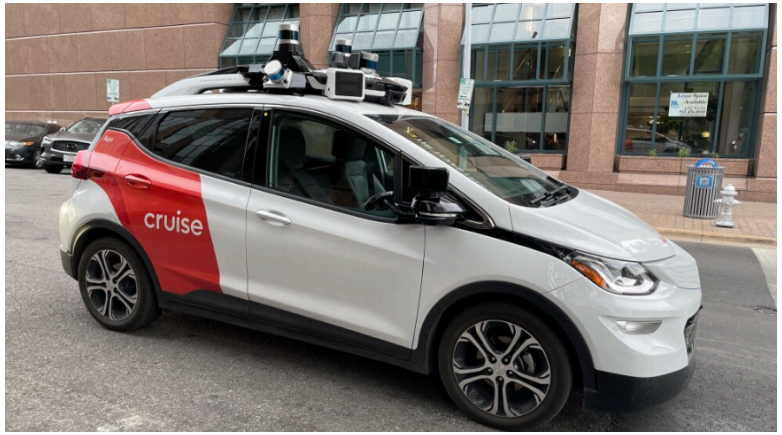
# PEAS example: automated taxi driver

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | Safe, fast, legal, comfortable trip, maximize profits | Roads, other traffic, pedestrians, customers | | |



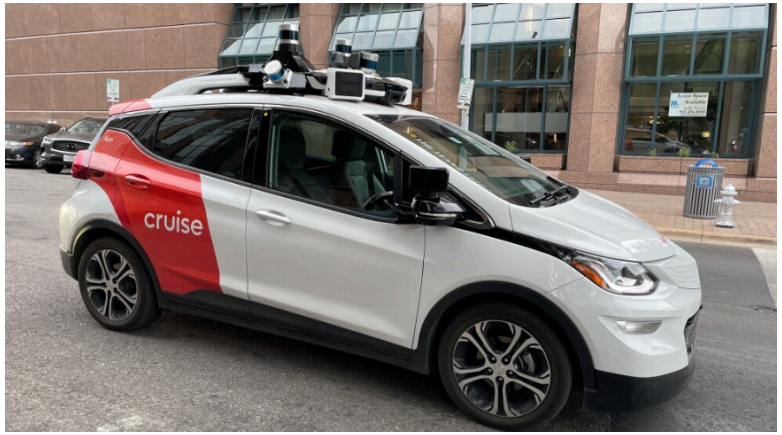
# PEAS example: automated taxi driver

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | Safe, fast, legal, comfortable trip, maximize profits | Roads, other traffic, pedestrians, customers | Steering, accelerator, brake, signal, horn, display | |



# PEAS example: automated taxi driver

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | Safe, fast, legal, comfortable trip, maximize profits | Roads, other traffic, pedestrians, customers | Steering, accelerator, brake, signal, horn, display | Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard |



# Exercise: PEAS for part-picking robot from a conveyor belt?

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Part-picking robot | | | | |

# Solution: PEAS for part-picking robot from a conveyor belt?

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Part-picking robot | Percentage of parts in correct bins | Conveyor belt with parts; bins | Jointed arm and hand | Camera, joint angle sensors |

# PEAS Exercise

Do Task 1 from today's Group Activity

# Classifying task environments

- A wide range of task environments might arise in AI
- We can categorize task environments along several dimensions, which are conducive to appropriately designing various agents
- We will consider six such general categories

# Task environment dimension-1

- **Fully observable** vs. **partially observable**
  - Does the agent's sensors give it access to the complete state of the environment at each point in time?
  - If so, the task environment is **fully observable**; otherwise it is **partially observable**.

# Task environment dimension-2

- **Single-agent** vs. **multi-agent**
  - Number of agents involved in the environment hence the distinction is clear enough.
  - We can also consider competitive vs. cooperative multi-agent environments.

# Task environment dimension-3

- **Deterministic** vs. **stochastic**
  - Is the next state of the environment completely determined by the current state and the actions executed by the agent thus far?
  - If so, the task environment is **deterministic**; otherwise it is **stochastic**.

# Task environment dimension-4

- **Episodic** vs. **sequential**
  - Is the agent's experience divided into atomic episodes, wherein the agent receives a percept and then performs a single action? This is an **episodic** task environment. The choice of action in each episode depends only on the episode itself.
  - In a **sequential** task environment, the previous decisions can effect the current decision, which in turn can affect future decisions.

# Task environment dimension-5

- **Static** vs. **dynamic**
  - Can the environment change while the agent is deliberating?
  - If so, the environment is **dynamic**; otherwise it is static.
  - A **semi-dynamic** environment is one in which the environment does not change as time passes by, but the agent's performance score does.

# Task environment dimension-6

- **Discrete** vs. **continuous**
  - **Discrete** quantities can be counted by the whole numbers and do not continuously vary (unlike **continuous** quantities).
  - This distinction applies to
    - state of the environment
    - the way time is treated
    - the percepts and
    - actions of the agent

# Exercise: categorizing task environment

- According to the six types of task environment let's categorize automated taxi agents' task environment as follows:

| Task Environment | Observable | Agents | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|---|---|

- fully observable or partially observable?
- single-agent or multi-agent?
- deterministic or stochastic?
- episodic or sequential?
- static or dynamic?
- discrete or continuous?