



University of  
**Strathclyde**  
Science

# ***Lecture 2: Intelligent Agents***

Dr John Levine

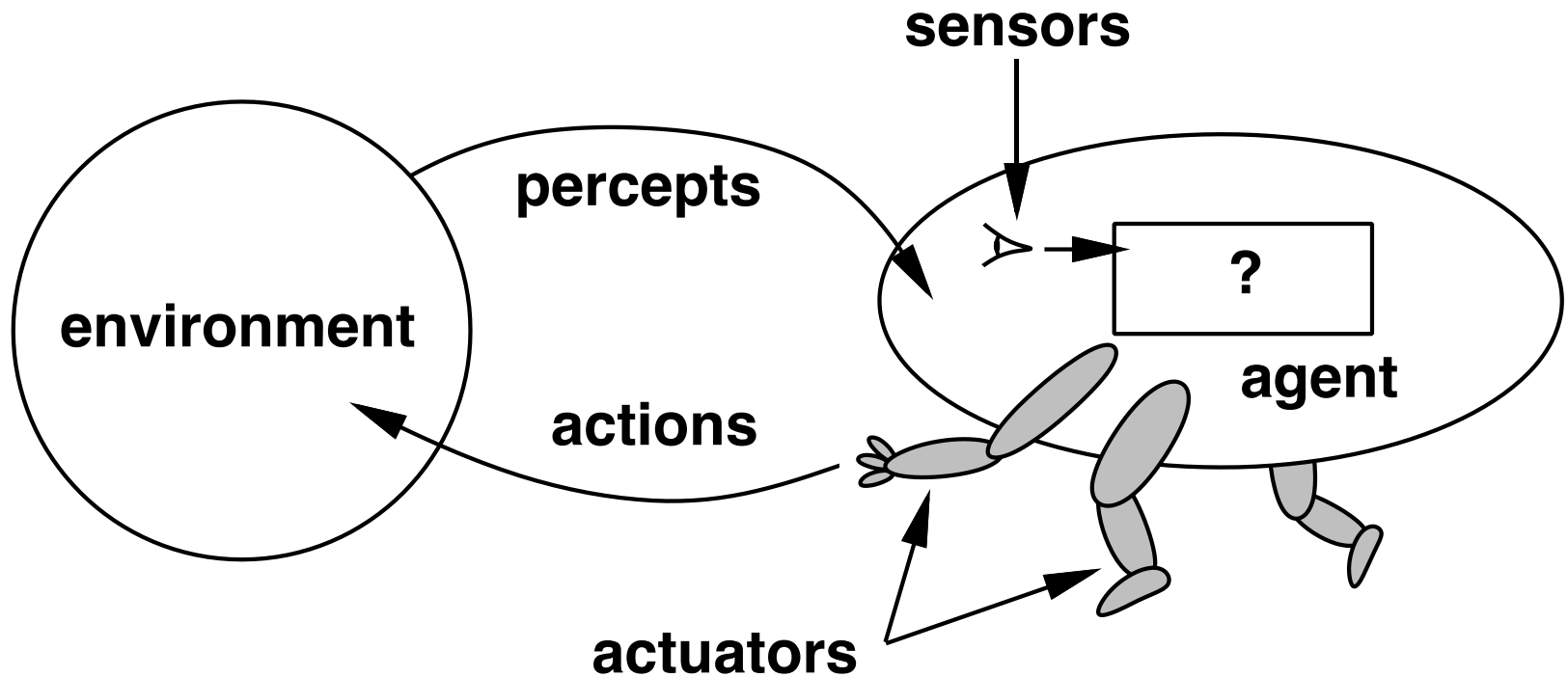
CS310 Foundations of Artificial Intelligence  
January 21st 2016

# Learning Outcomes

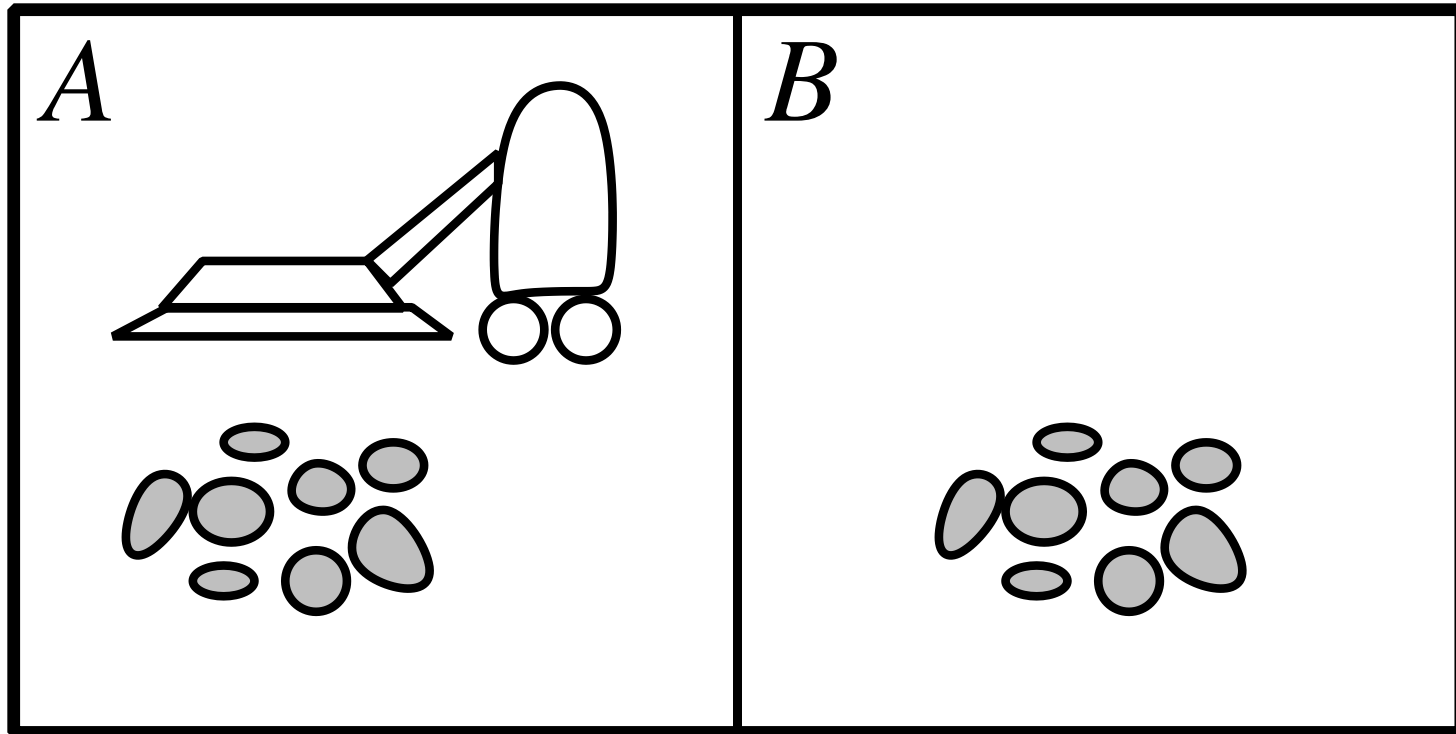
At the end of this lecture, you will know...

1. What an agent looks like
2. What is meant by “percept” and “percept sequence”
3. How to define an agent function and agent program
4. How to define properties of agents using PEAS
5. How to define properties of environments
6. The difference between a reactive (reflex) agent and a deliberative (goal/utility-based) agent

# What does an agent look like?



# Example: A Robot Vacuum Cleaner



# Percepts and Percept Sequences

- A *percept* is a complete set of readings from all of the agent's sensors at an instant in time
- For the robot vacuum cleaner, this will consist of its location and whether the floor is clean or dirty
- Example percept: [A, dirty]
- A *percept sequence* is a complete ordered list of the percepts that the agent received since time began
- Example: [ [A, dirty], [A, dirty], [A, clean], [B, dirty], ... ]

# An Agent Function

- An *agent function* is a theoretical device which maps from any possible percept sequence to an *action*:

Percept sequence	Action
$[A, \textit{Clean}]$	$\textit{Right}$
$[A, \textit{Dirty}]$	$\textit{Suck}$
$[B, \textit{Clean}]$	$\textit{Left}$
$[B, \textit{Dirty}]$	$\textit{Suck}$
$[A, \textit{Clean}], [A, \textit{Clean}]$	$\textit{Right}$
$[A, \textit{Clean}], [A, \textit{Dirty}]$	$\textit{Suck}$
⋮	⋮

# An Agent Program

- An *agent program* is what we run on the architecture of the agent to implement the 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
```

- What is the *right* agent function?
- Can it be implemented efficiently on the architecture?

# Evaluating the agent: performance

- How can we evaluate an agent, such as our robotic vacuum cleaner?
- Look at its *performance* on the task in hand
- In other words, give it a score: how would you do this for the vacuum cleaner?
- The more intelligent the agent, the higher its score
- Simple agents can score well at simple tasks, but for complex tasks we need more sophisticated agents: rational agents which *reason* to achieve a high score



# PEAS

- To design a rational agent, we must specify the “task environment” for the agent
- We use the acronym PEAS for this:
  - **P**erformance: how well does the agent do?
  - **E**nvironment: what does the agent’s environment look like?
  - **A**ctuators: what actuators does the agent have to perform its actions with?
  - **S**ensors: what sensors does the agent have to perceive the environment with?

# PEAS Example

- Example: a Lego Mindstorms robot which can move around safely in a cluttered room:



# Properties of environments

- A unknown and cluttered room with other agents in it is a more difficult place to live in than an empty room
- A simple task in a simple environment means that we can get top performance out of a simple agent
- A complex task in a complex environment requires a very sophisticated agent
- Consider the task and environment for an automated taxi driver!

# Dimensions of the environment

- We use six dimensions to define the environment:
  - Fully observable or partially observable?
  - Single agent or multi-agent?
  - Deterministic or stochastic?
  - Episodic or sequential?
  - Static or dynamic?
  - Discrete or continuous?

# Dimensions of the environment

	Solitaire	Backgammon	Internet shopping	Taxi
<u>Fully observable?</u>	No	Yes	No	No
<u>Deterministic?</u>	Yes*	No	Partly	No
<u>Episodic?</u>	No	No	No	No
<u>Static?</u>	Yes	Semi	Semi	No
<u>Discrete?</u>	Yes	Yes	Yes	No
<u>Single-agent?</u>	Yes	No	No	No

\*After the cards have been dealt

**The environment type largely determines the agent design**

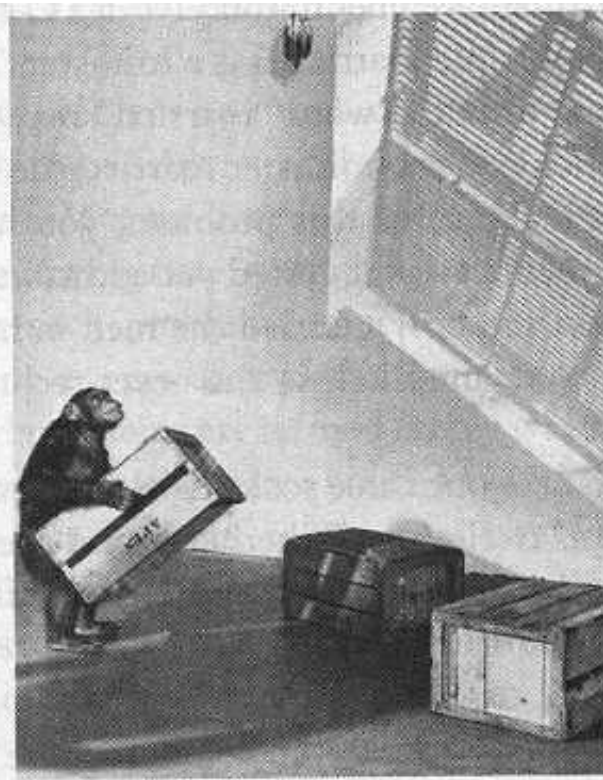
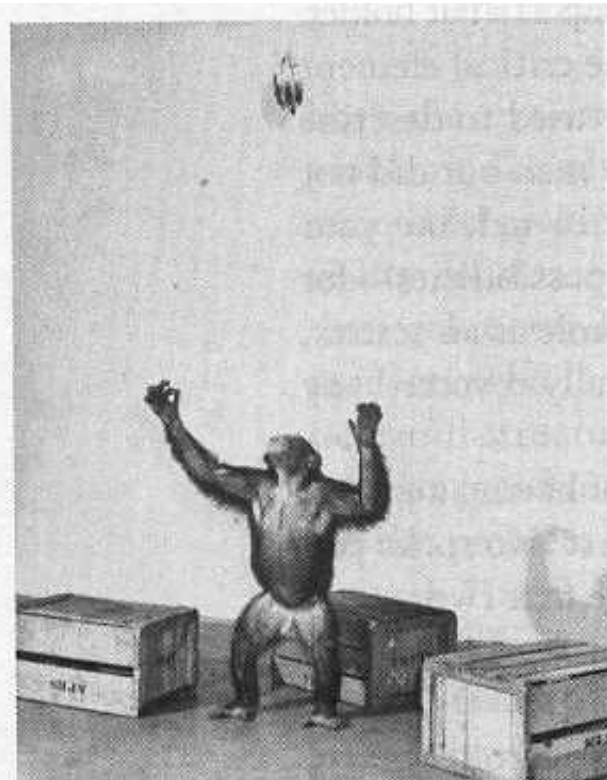
Real world: partially observable, stochastic, sequential, dynamic, continuous, multi-agent

# Reactive Agents

- If an agent only “reacts” to its current percept, we call it a reactive (or reflex) agent
- Actions are chosen rules of the form “if condition then action” (or something equivalent to this)
- Examples: the simple vacuum cleaner controller, the hand-coded and evolved agents in EvoTanks
- Good performance only at simple tasks - would need an unfeasibly large number of rules for a complicated task, such as playing chess



# Deliberative Agents



# Deliberative Agents

- Deliberation is the process of thinking about how your actions will affect the world before committing to it
- Given a choice of actions, choose the one that gets you closer to your goals, or closer to a state of high utility
- Since the goal or state of high utility may be some way off, need to think about sequences of actions
- This leads to a *search problem...*





# What happens next?

- Lecture 3 on Tuesday at 1pm: “How to Search Huge Graphs”
- Tutorial 1: getting to grips with AI agents, PEAS and environments (for Tuesday 26th, 2pm)
- Practical work and labs will start in Week 3