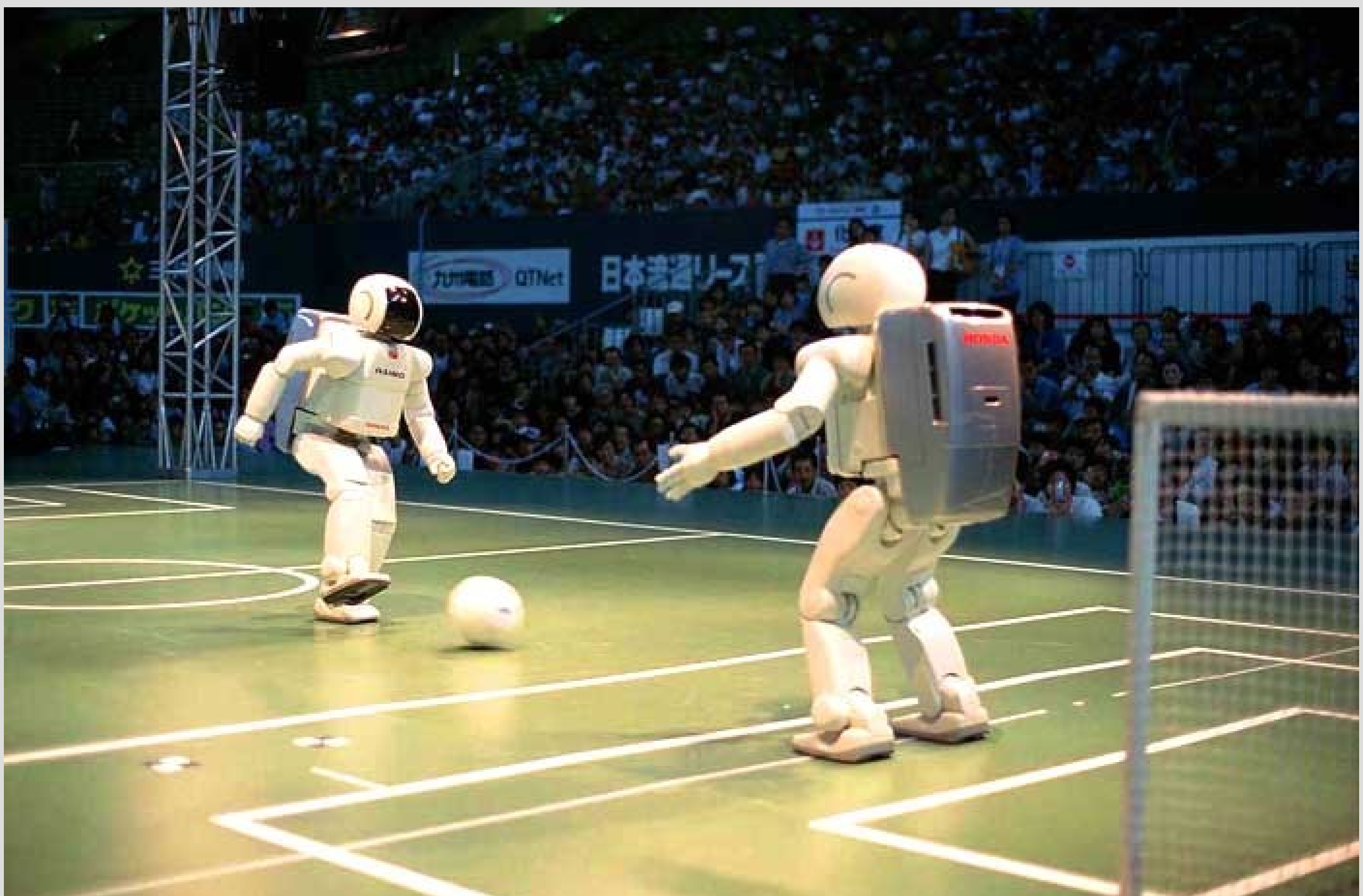
Rational Agents (Ch. 2)



Extra credit!

Occasionally we will have in-class activities for extra credit (+3%)

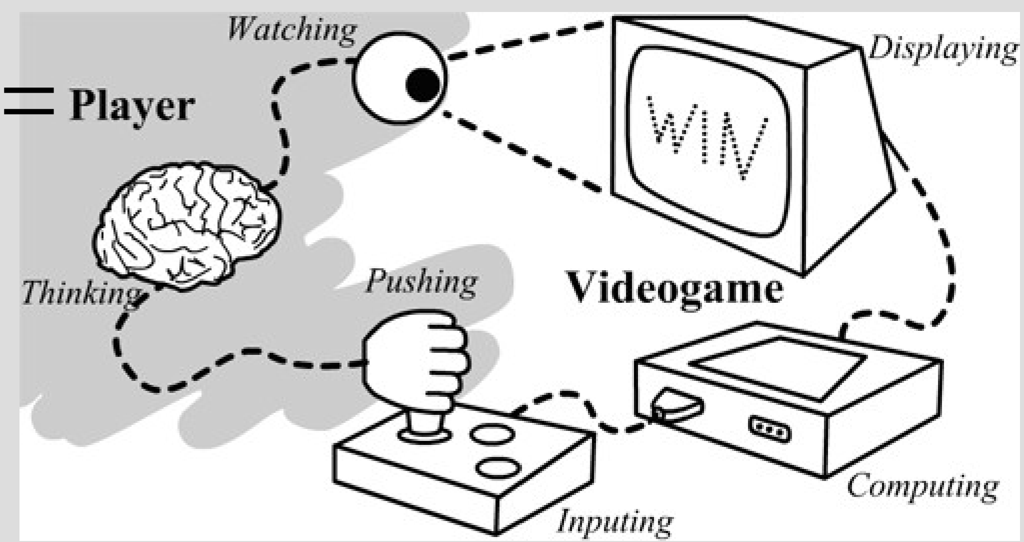
You do not need to have a full or correct answer to get credit, but you do need to attempt the problem (and show work)

Rational agent

An agent/robot must be able to perceive and interact with the environment

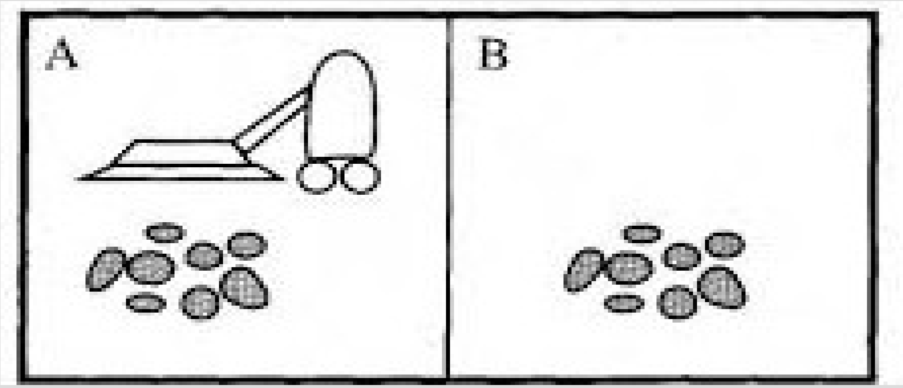
A rational agent is one that always takes the best action (possibly expected best)

Agent =



Rational agent

Consider the case of a simple vacuum agent



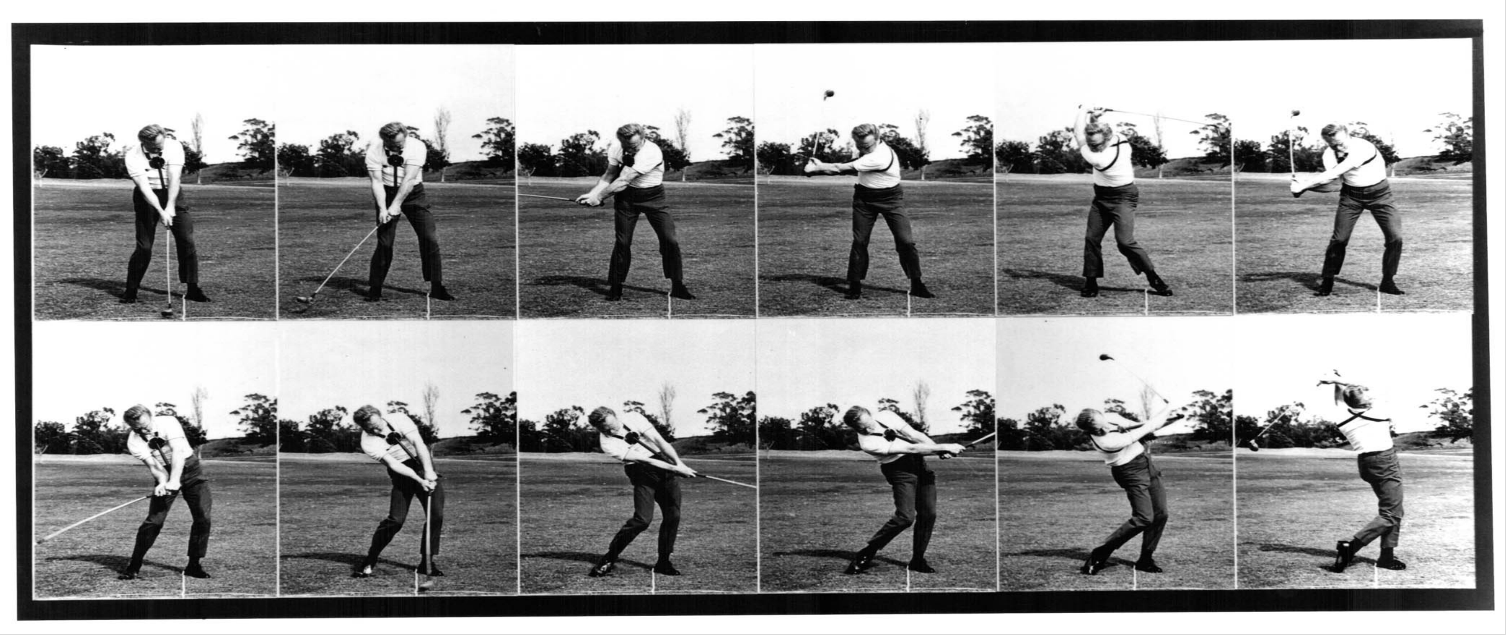
Environment: [state A] and [state B], both possibly with dirt that does not respawn

Actions: [move left], [move right] or [suck]

Perception: current location, [dirty or clean]

Rational agent

An agent's <u>percept</u> is the sequence of perceptions that it has seen up to this point



For the vacuum agent, one percept might be:
[A, Dirty], [A, Clean], [B, Dirty]

Rational agent

There are two ways to describe an agent's action using the percept:

1. Agent function = directly map a percept to action
2. Agent program = logic dictating next action (percept as an input to logic)

The agent function is basically a look-up table, and is typically much larger

Rational agent

An agent function for vacuum agent:

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$

A corresponding agent program:

- if [Dirty], return [Suck]
- if at [state A], return [move right]
- if at [state B], return [move left]

Rational agent

In order to determine if the vacuum agent is rational I need a performance measure

Under which of these metrics is the agent program on the previous slide rational?

1. Have a clean floor in A and B
2. Have a clean floor as fast as possible
3. Have a clean floor with moving as little as possible
4. Maximize the amount of time sucking

Rational agent

You want to express the performance measure in terms of the environment not the agent

For example, if we describe a measure as: "Suck up the most dirt"

A rational vacuum agent would suck up dirt then dump it back to be sucked up again...

This will not lead to a clean floor

Rational agent

Performance measure: “-50 points per time step a state is dirty and -1 point per move”

Is our agent rational (with the proposed agent program) if...

1. Dirt does not reappear
2. Dirt always reappears the next time step
3. Dirt has a 30% chance of reappearing
4. Dirt reappears but at an unknown rate

Rational agent

If we do not know how often dirt will reappear, a rational agent might need to learn

Learning can use prior knowledge to estimate how often dirt tends to reappear, but should value actual observations more (its percept)

The agent might need to explore and take sub-optimal short-term actions to find a better long-term solution

Rational agent

To recap, a rational agent depends on:

1. Performance measure
2. Prior knowledge of the environment
3. Actions available
4. Percept to current time

You need to know all of these before you can determine rationality

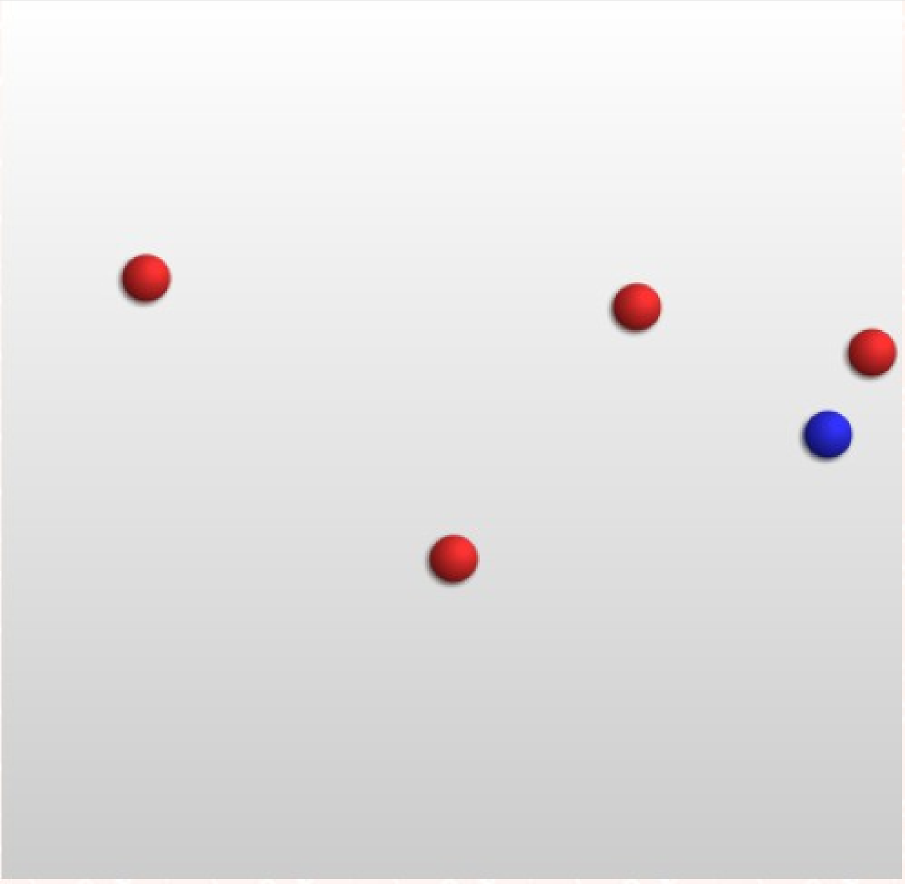
Rational agent

To fully specify the situation we need to know about the task environment too (abbreviated PEAS)

Performance measure
Environment
Actuators
Sensors

Rational agent

Particle game:
http://www.ragdollsoft.com/particles/



Rational agent

Agent type	Performance	Environment	Actuators	Sensors
Vacuum	time to clean	A, B, dirt	suck, move	dust sensor
Student	GPA, honors	campus, dorm	do HW, take test	eye, ear, hand
Particles	time alive	boarder, red balls	move mouse	screen-shot

Environment classification

Environments can be further classified on the following characteristics:(right side harder)

1. Fully vs. partially observable
2. Single vs. multi-agent
3. Deterministic vs. stochastic
4. Episodic vs. sequential
5. Static vs. dynamic
6. Discrete vs. continuous
7. Known vs. unknown

Environment classification

In a fully observable environment, agents can see every part.

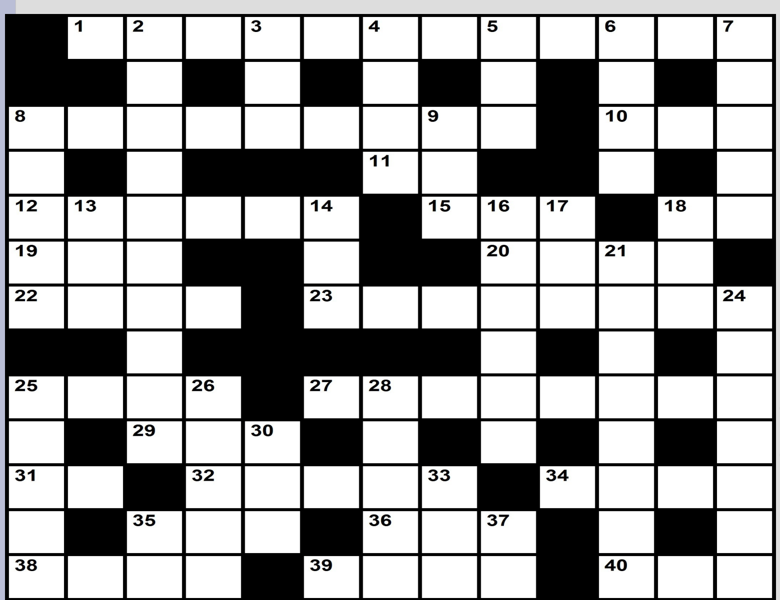
Agents can only see part of the environment if it is partially observable

Full ↗ Partial →

Environment classification

If your agent is the only one, the environment is a <u>single agent</u> environment

More than one is a <u>multi-agent</u> environment (possibly cooperative or competitive)



← single

multi →

Environment classification

If your state+action has a known effect in the environment, it is <u>deterministic</u>

If actions have a distribution (probability) of possible effects, it is <u>stochastic</u>

deterministic

stochastic

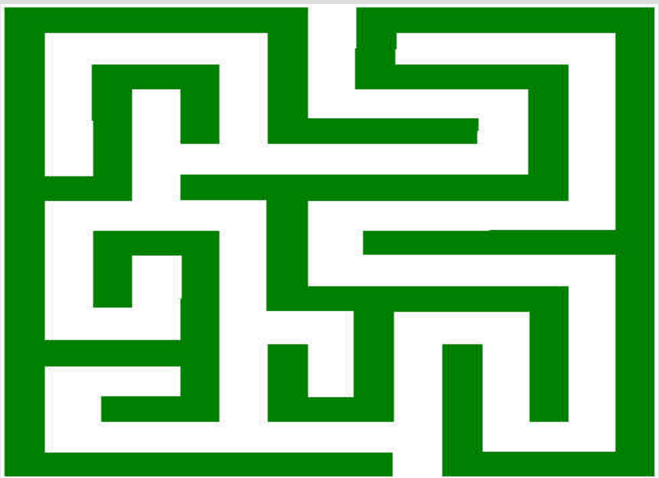
Environment classification

An episodic environment is where the previous action does not effect the next observation (i.e. independent)

If there is the next action depends on the previous, the environment is sequential

← episodic

sequential →



Environment classification

If the environment only changes when you make an action, it is static

a dynamic environment can change while your agent is thinking or observing

static

dynamic

Environment classification

Discrete = separate/distinct (events)
Continuous = fluid transition (between events)

This classification can applies: agent's percept and actions, environment's time and states

discrete (state)

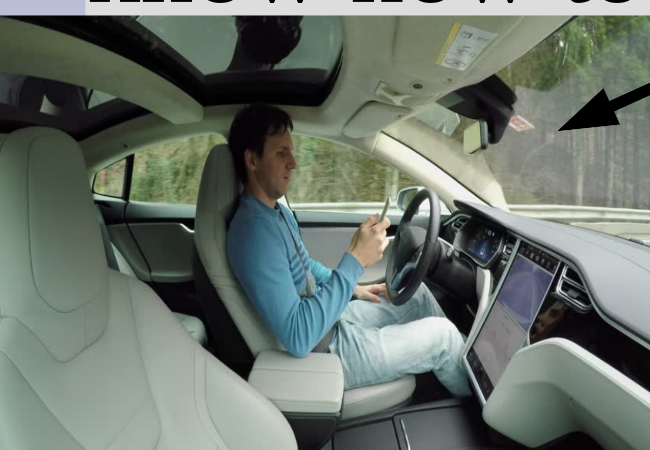
continuous (state)

Environment classification

<u>Known</u> = agent's actions have known effects on the environment

<u>Unknown</u> = the actions have an initially unknown effect on the environment (can learn)

know how to stop



do not know how to stop

Environment classification

Pick a game/hobby/sport/pastime/whatever and describe both the PEAS and whether the environment/agent is:

1. Fully vs. partially observable
2. Single vs. multi-agent
3. Deterministic vs. stochastic
4. Episodic vs. sequential
5. Static vs. dynamic
6. Discrete vs. continuous
7. Known vs. unknown

Environment classification

Agent type	Perfor mance	Environ ment	Actuator s	Sensors
Particles	time alive	boarder, red balls	move mouse	screen-shot

Fully observable, single agent, deterministic, sequential, dynamic,
continuous (time, state, action, and percept),
known (to me!)

Agent models

Can also classify agents into four categories:

1. Simple reflex
2. Model-based reflex
3. Goal based
4. Utility based

Top is typically simpler and harder to adapt to similar problems, while bottom is more general representations

Agent models

A <u>simple reflex</u> agents acts only on the most recent part of the percept and not the whole history

Our vacuum agent is of this type, as it only looks at the current state and not any previous

These can be generalized as:
“if state = _____ then do action _____”
(often can fail or loop infinitely)

Agent models

A <u>model-based reflex</u> agent needs to have a representation of the environment in memory (called <u>internal state</u>)

This internal state is updated with each observation and then dictates actions

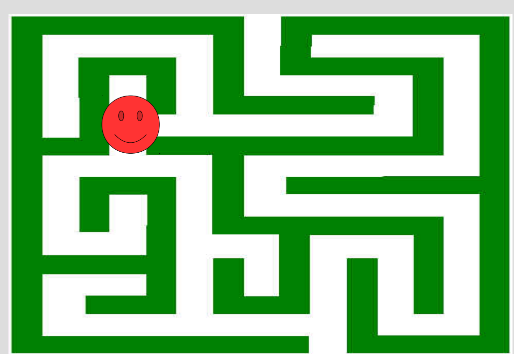
The degree that the environment is modeled is up to the agent/designer (a single bit vs. a full representation)

Agent models

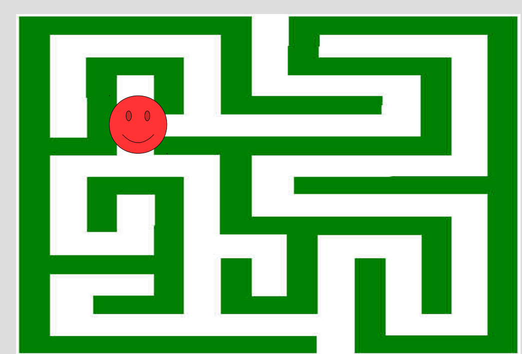
This internal state should be from the agent's perspective, not a global perspective
(as same global state might have different actions)

Consider these pictures of a maze:
Which way to go?

Pic 1



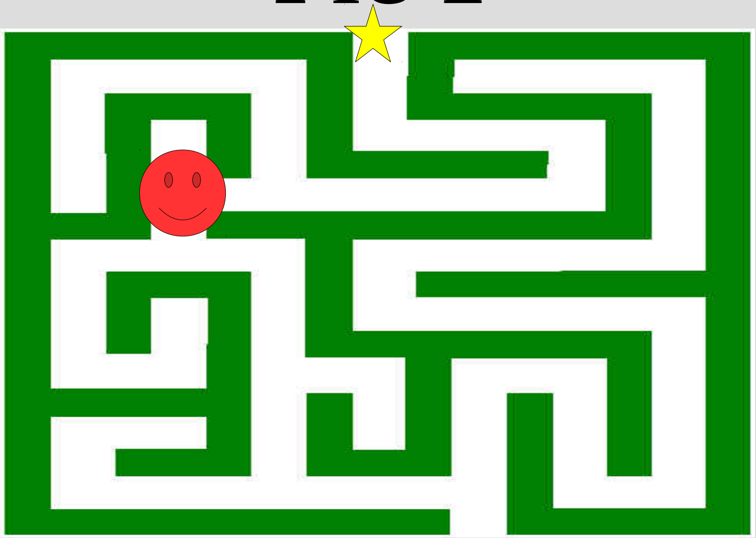
Pic 2



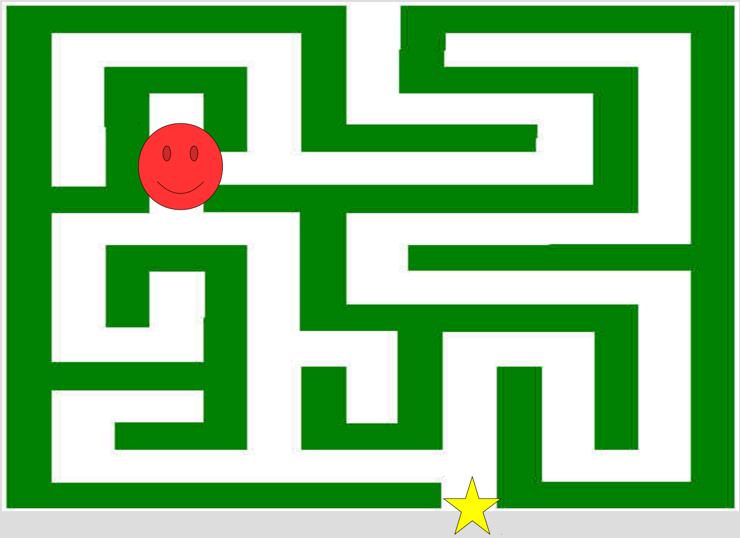
Agent models

The global perspective is the same, but the agents could have different goals (stars)

Pic 1



Pic 2



Goals are not global information

Agent models

For the vacuum agent if the dirt does not reappear, then we do not want to keep moving

The simple reflex agent program cannot do this, so we would have to have some memory (or model)

This could be as simple as a flag indicating whether or not we have checked the other state

Agent models

The <u>goal based</u> agent is more general than the model-based agent

In addition to the environment model, it has a <u>goal</u> indicating a desired configuration

Abstracting to a goals generalizes your method to different (similar) problems
(for example, a model-based agent could solve one maze, but a goal can solve any maze)

Agent models

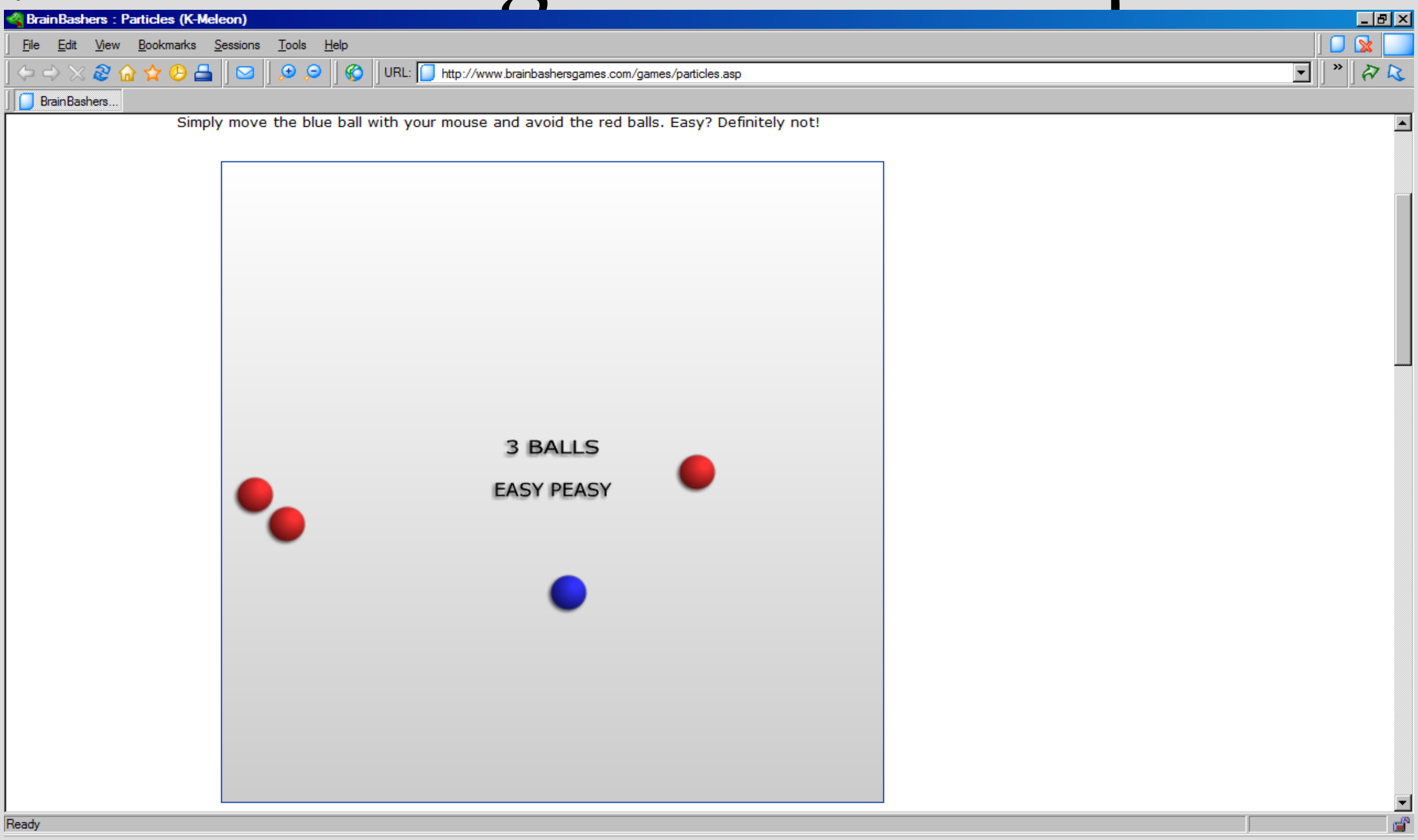
A utility based agent maps the sequence of states (or actions) to a real value

Goals can describe general terms as “success” or “failure”, but there is no degree of success

In the maze example, a goal based agent can find the exit. But a utility based agent can find the shortest path to the exit

Agent models

What is the agent model of particles?



Think of a way to improve the agent and describe what model it is now

Agent learning

For many complicated problems (facial recognition, high degree of freedom robot movement), it would be too hard to explicitly tell the agent what to do

Instead, we build a framework to learn the problem and let the agent decide what to do

This is less work and allows the agent to adapt if the environment changes

Agent learning

There are four main components to learning:

1. Critic = evaluates how well the agent is doing and whether it needs to change actions (similar to performance measure)
2. Learning element = incorporate new information to improve agent
3. Performance element = selects action agent will do (exploit known best solution)
4. Problem generator = find new solutions (explore problem space for better solution)

State structure

States can be generalized into three categories:

1. Atomic (Ch. 3-5, 15, 17)
2. Factored (Ch. 6-7, 10-11, 13-16, 18, 20-21)
3. Structured (Ch. 8-9, 12, 14, 19, 22-23)

(Top are simpler, bottom are more general)

Occam's razor = if two results are identical, use the simpler approach

State structure

An <u>atomic</u> state has no sub-parts and acts as a simple unique identifier

An example is an elevator:
Elevator = agent (actions = up/down)
Floor = state

In this example, when someone requests the elevator on floor 7, the only information the agent has is what floor it currently is on

State structure

Another example of an atomic representation is simple path finding:
If we start (here) in Amundson B75, how would you get to Keller's CS office?

Am. B75 -> Hallway1 -> Tunnel -> Hallway2 -> Elevator -> Hallway3 -> CS office

The words above hold no special meaning other than differentiating from each other

State structure

A factored state has a fixed number of variables/attributes associated with it

Our simple vacuum example is factored, as each state has an id (A or B) along with a "dirty" property

In particles, each state has a set of red balls with locations along with the blue ball location

State structure

Structured states simply describe objects and their relationship to others

Suppose we have 3 blocks: A, B and C
We could describe: A on top of B, C next to B

A factored representation would have to enumerate all possible configurations of A, B and C to be as representative

State structure

We will start using <u>structured</u> approaches when we deal with logic:

Summer implies Warm
Warm implies T-Shirt

The current state might be:
!Summer (¬Summer)
but the states have intrinsic relations between each other (not just actions)