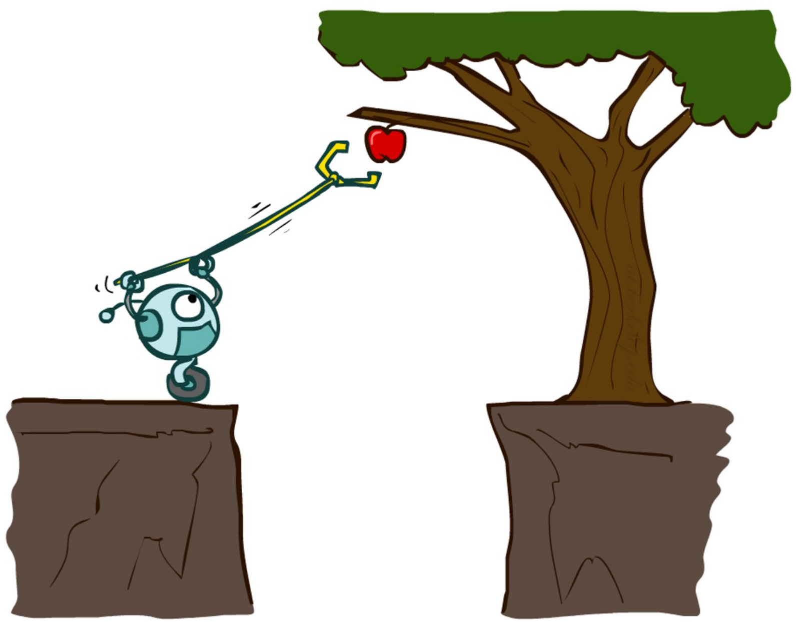
Agents and Environments

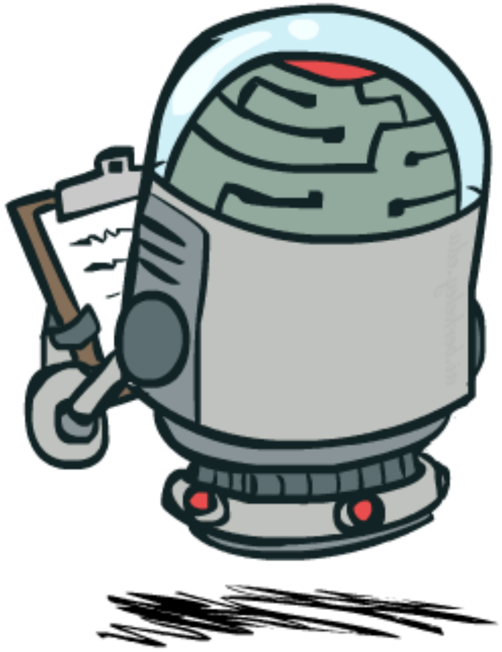
CSE 473: Introduction to Artificial Intelligence



James Weichert (jpw@cs)
cs.uw.edu/473
acknowledgements

Welcome!
Favorite fruit?
sli.do **#cse473**

Administrivia



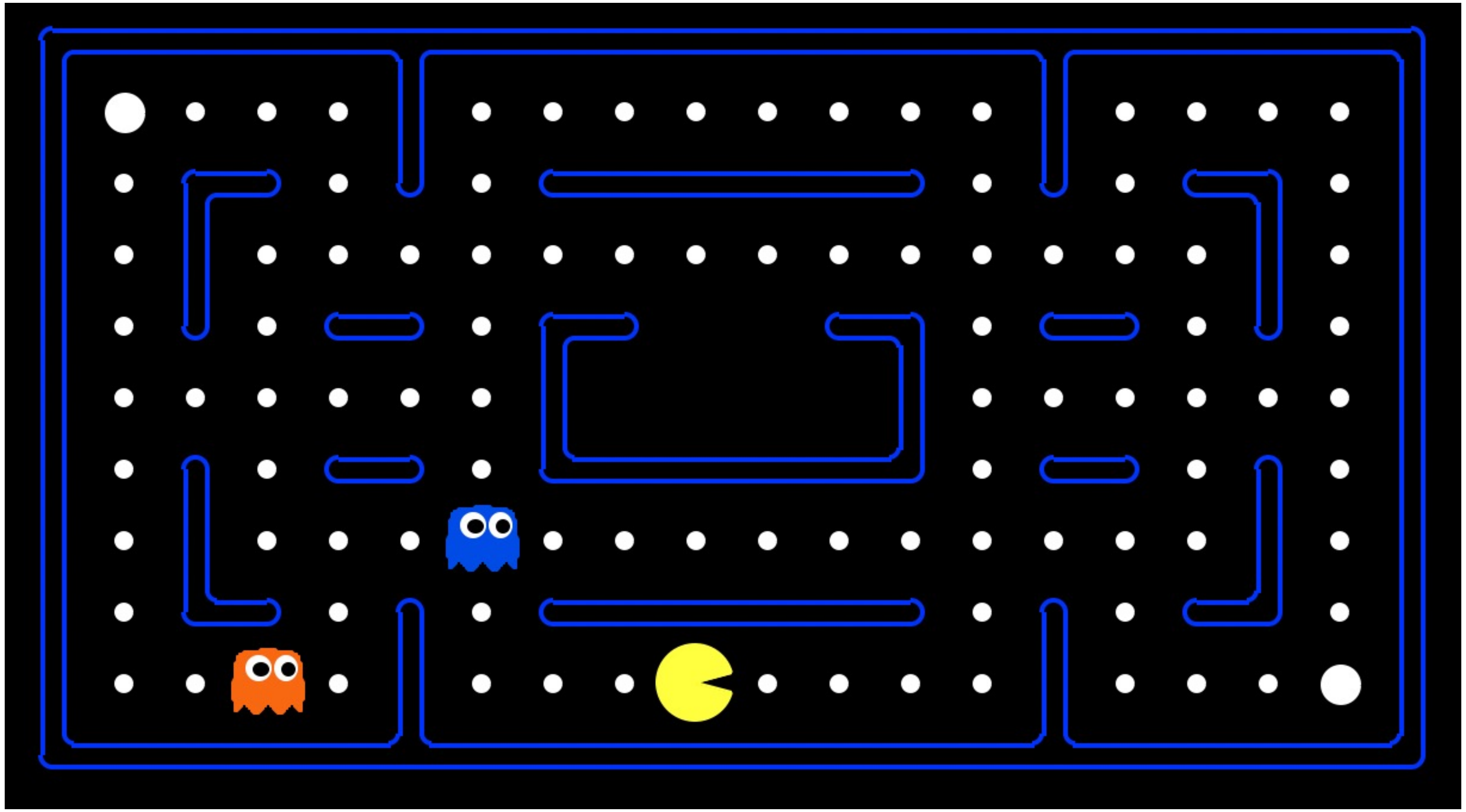
ANNOUNCEMENTS

- Make sure you can access Ed, Gradescope, Panopto
- Remember to complete the **practice problems** for each lecture

ASSIGNMENTS

- **Project 0** (*optional*) is out!
  - **due** Thursday (6.25)
- **Homework 1** released Friday
  - **due** next Thursday (7.2)
- **Project 1** released Friday
  - **due** in two weeks (7.9)

How should we model the world?



Modeling Pacman

(WORKING) DEFINITION

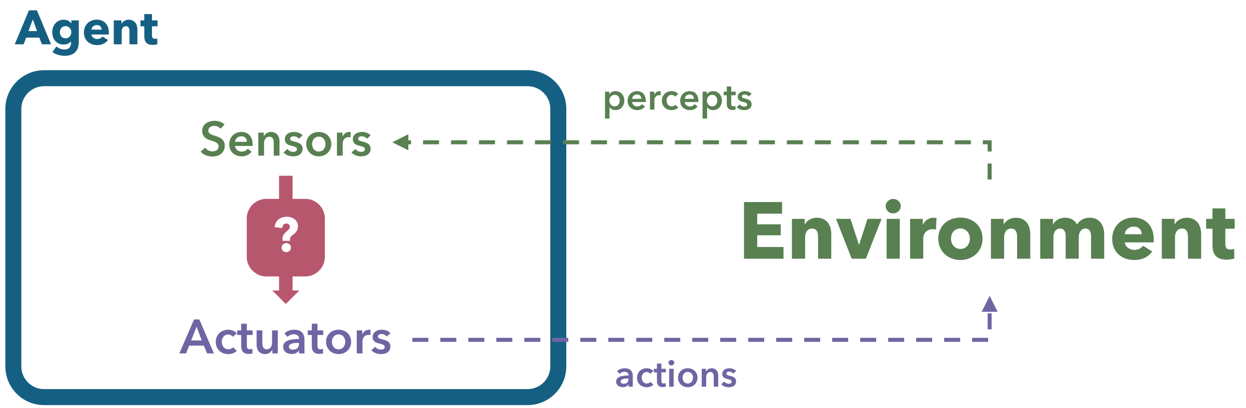
AI comprises computer systems that demonstrate the ability to **achieve specified objectives** through **independent action**.

Agents

FOUNDATIONS

One approach is to model the interactions of individual **agents**

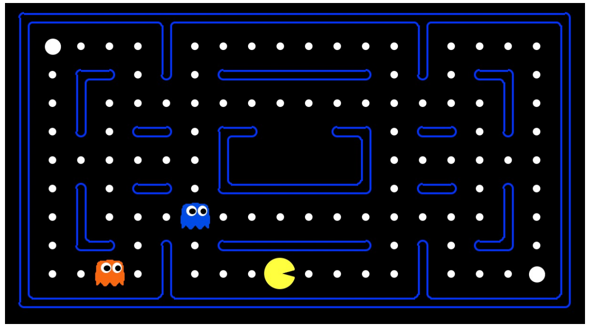
An **agent** *perceives* its environment through **sensors** and *acts* on it through **actuators**.



Agents?

EXAMPLE

Do the following fit our definition?



- **Pacman navigating a maze** ✅
  - **Sensors**: position, position of dots and ghosts
  - **Actuators**: NSEW movements (really, directions)

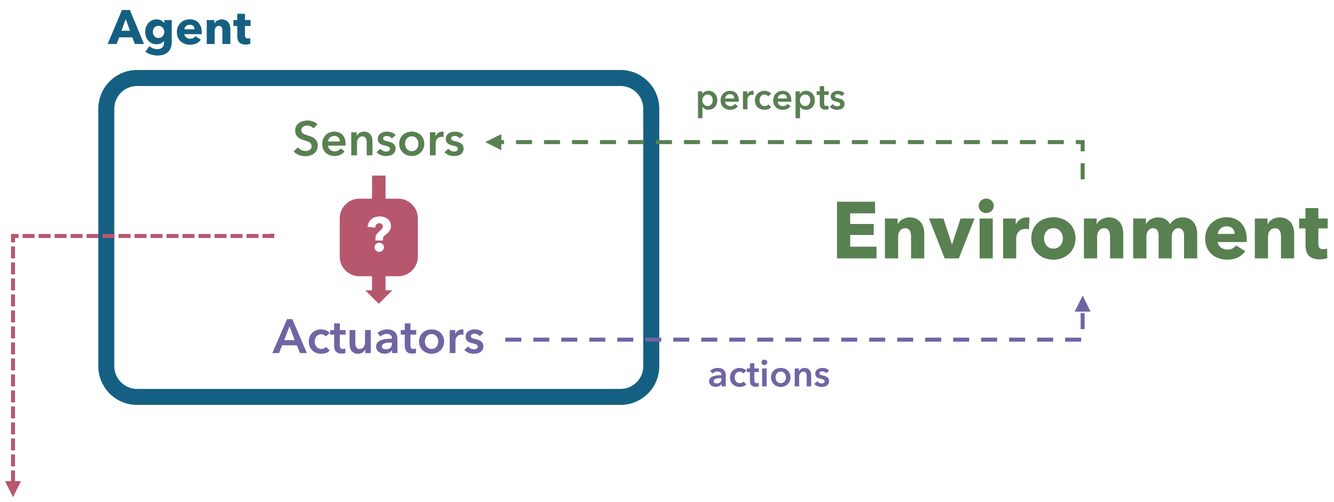
- **Humans** ✅
  - **Sensors**: vision, sound, touch, smell, taste
  - **Actuators**: muscles, secretions, brain state

- Pocket **calculators** ✅
  - **Sensors**: key press sensors, light sensor
  - **Actuators**: digit display

Now What?

FOUNDATIONS

- We've modeled the (simplified) world. So we're done, right?



Finding the **agent function** that maps percepts to actions is *very* hard.

This is ***the*** **challenge for artificial intelligence.**

Agent Functions and Programs

DEFINITION

- An **agent function** defines the relationship between percepts and actions

$$f: P^* \rightarrow A$$

- An **agent program** is an (approximate) implementation of that function

$$f \approx \text{Agent}(\text{program}, \text{machine})$$

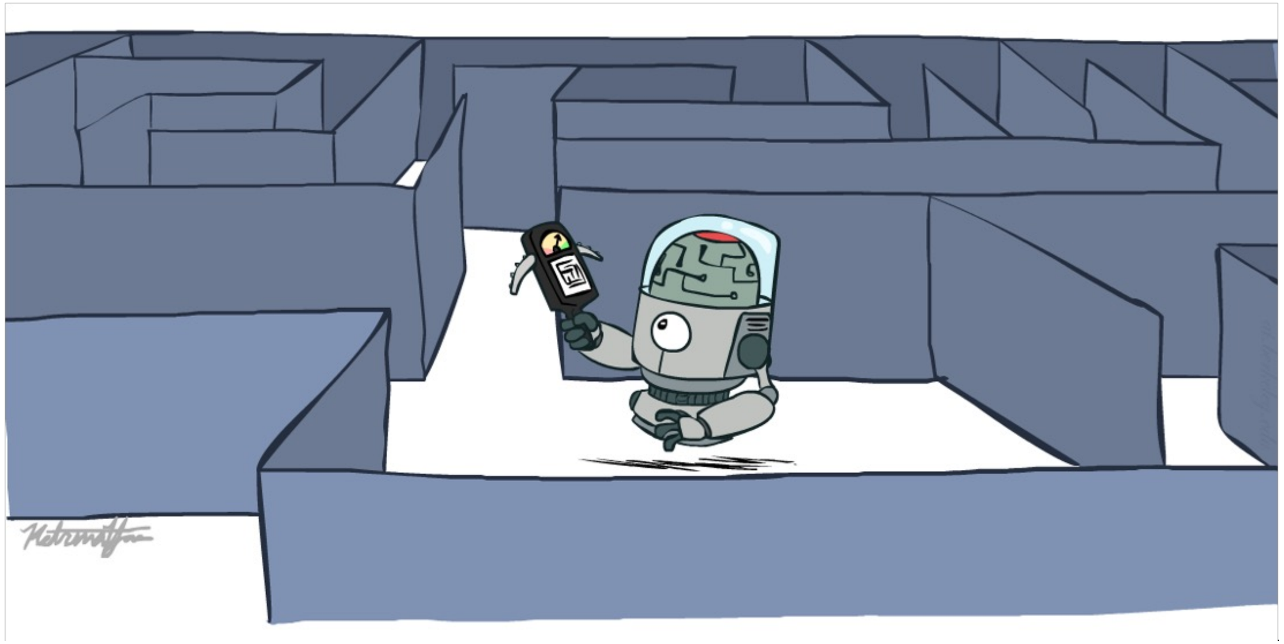
Questions?

live and on sli.do #cse473

Rationality

DEFINITION

A **rational** agent selects actions that maximize its *expected* **utility**



Reflex Agents

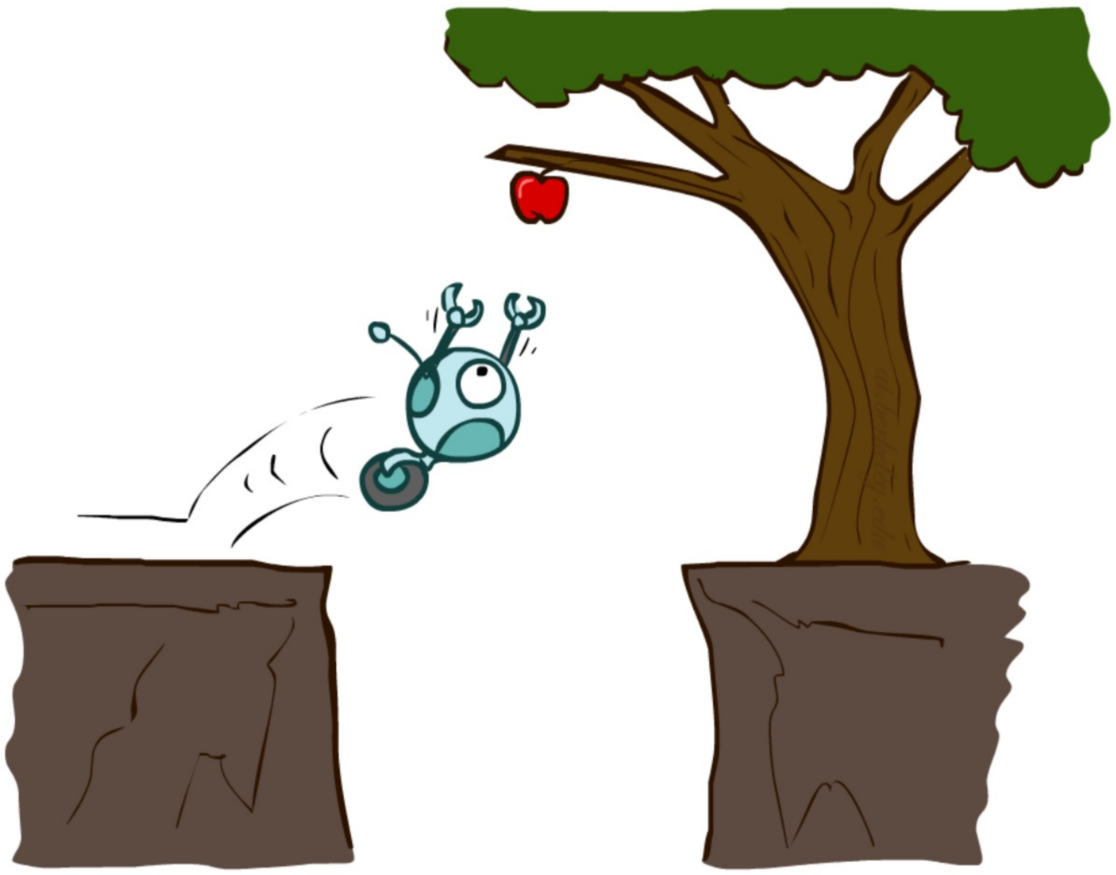
EXAMPLE

Reflex agents consider how the world *is*, without planning

- Choose action based on current percept
- May have memory or model of world's current state
- Do not consider future consequences of their actions

Can reflex agents be rational?

Yes!



Planning Agents

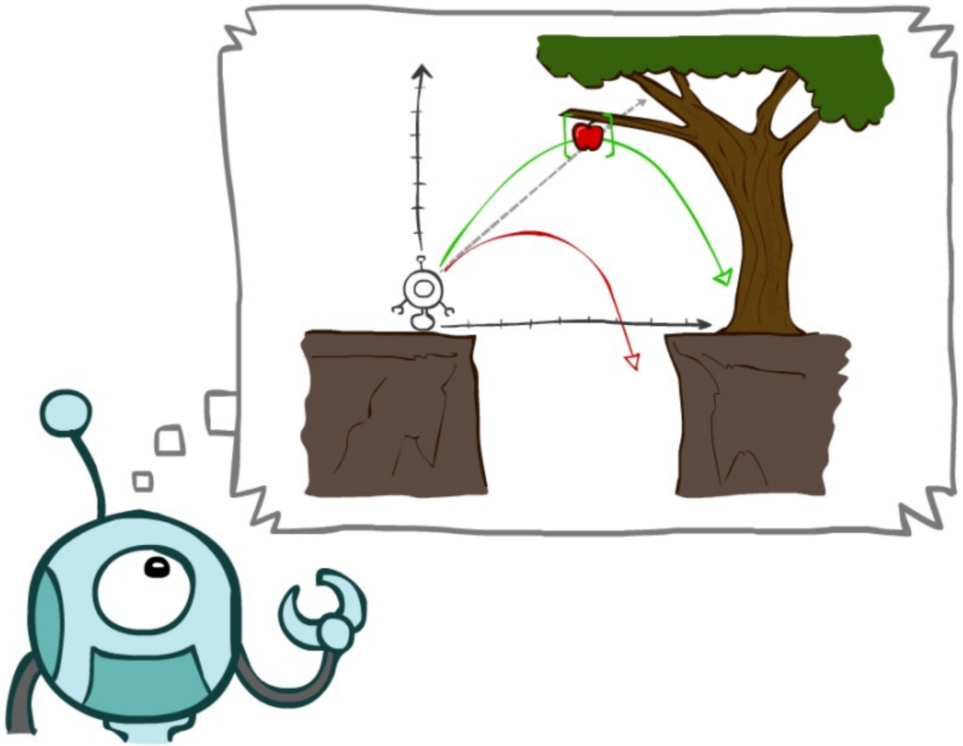
EXAMPLE

Planning agents **forecast** to consider how the world *would* be

- Choose action based on (hypothesized) consequences of actions
- *Must* have a model of the world and how it evolves in response to actions
- *Must* formulate and check for a goal

Why not just plan everything?

Randomness, Complexity



Vacuum World

EXAMPLE

What forms could this vacuuming agent take? What could go wrong?

Reflex Agent

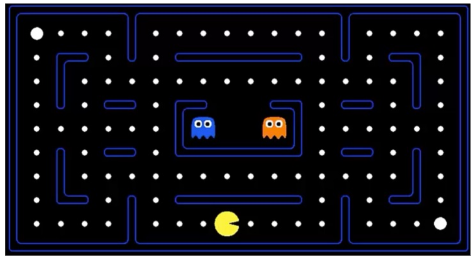
Planning Agent

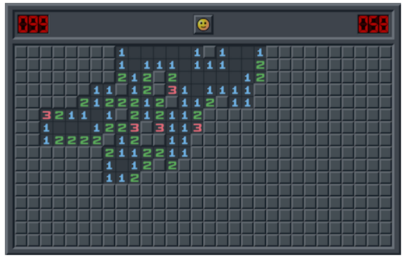
Environments

FOUNDATIONS

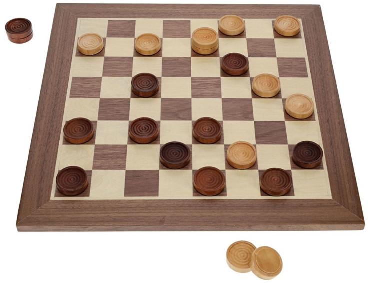
What kind of environment is the agent dealing with?

fully observable *vs.* partially observable

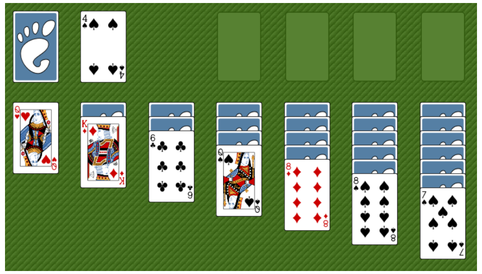




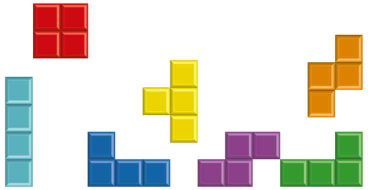
deterministic *vs.* stochastic



single agent *vs.* multi-agent



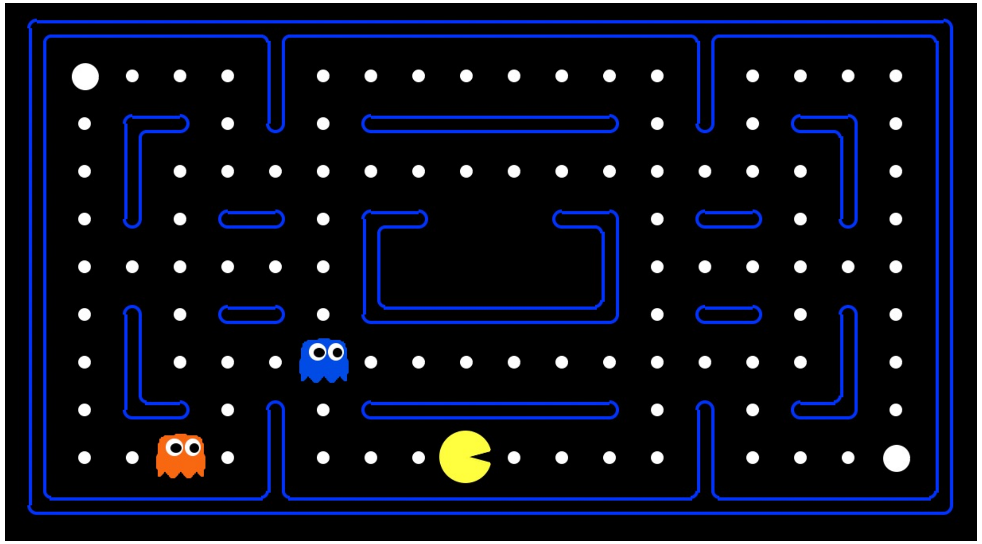
static *vs.* sequential



Pacman Environment

EXAMPLE

- **fully observable** vs. partially observable
- **deterministic** vs. stochastic
- single agent vs. multi-agent
    - Project 1: single agent (maze navigation)
    - Project 2: multi-agent (Pacman + ghosts)
- static vs. **sequential**



Questions?

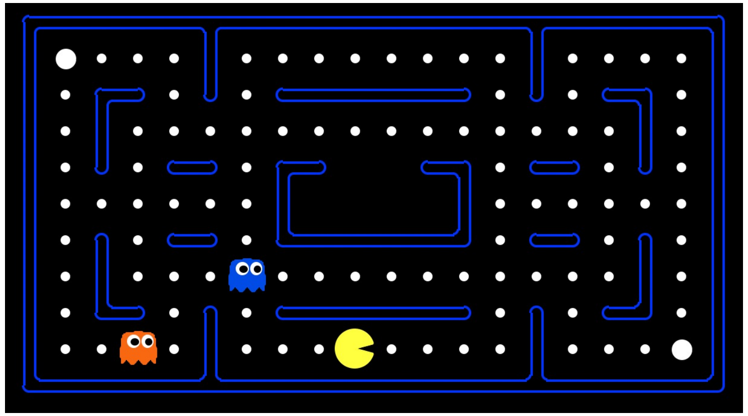
live and on sli.do #cse473

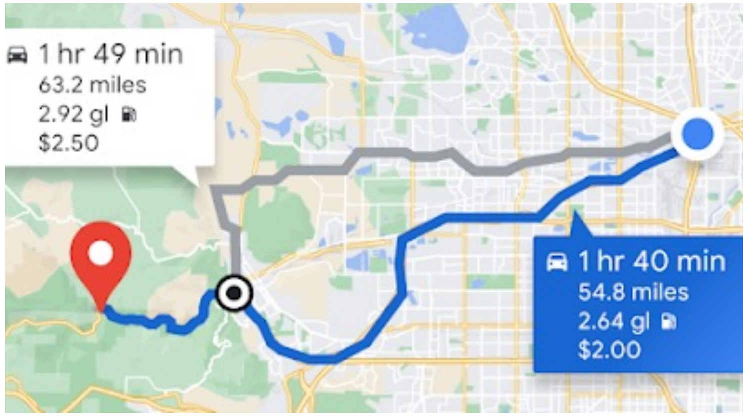
Search

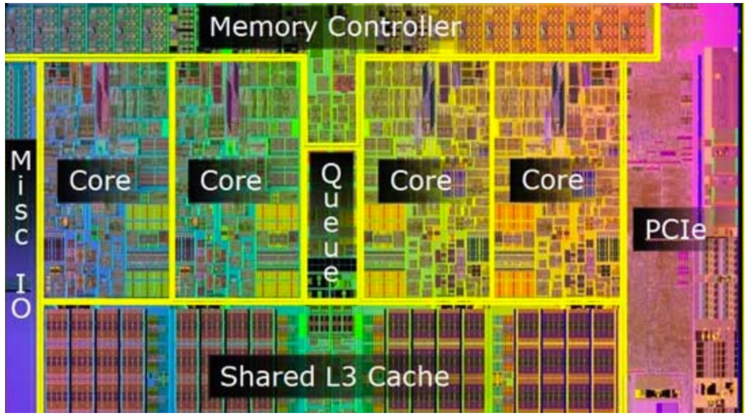
FOUNDATIONS

Search: It's Not Just For Mazes™

EXAMPLE







Search Problems

DEFINITION

A **search problem** consists of:

- **state space**

  S
- **actions** in each state

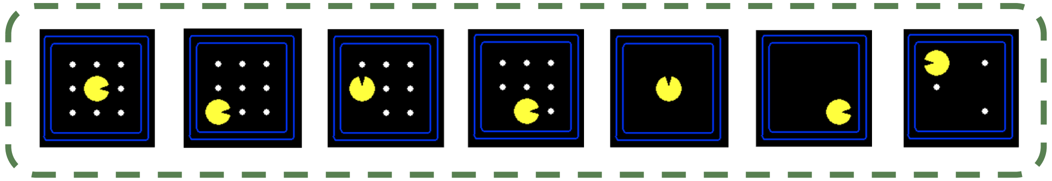
  $a \in A(s)$
- **successor function**

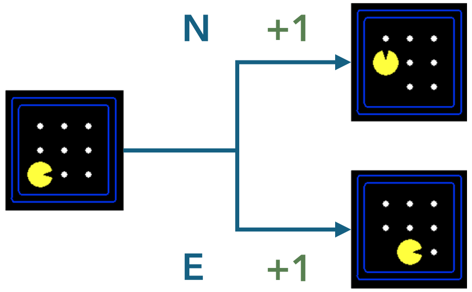
  $s' = T(s, a)$
- action **cost** (reward model)

  $R(s, a, s')$
- **start state**

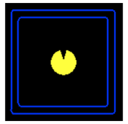
  $s_0 \in S$
- **goal test**

  $G(s)$





G() = **false**

G() = **true**

Search Problems Solutions

DEFINITION

A **search problem** consists of:

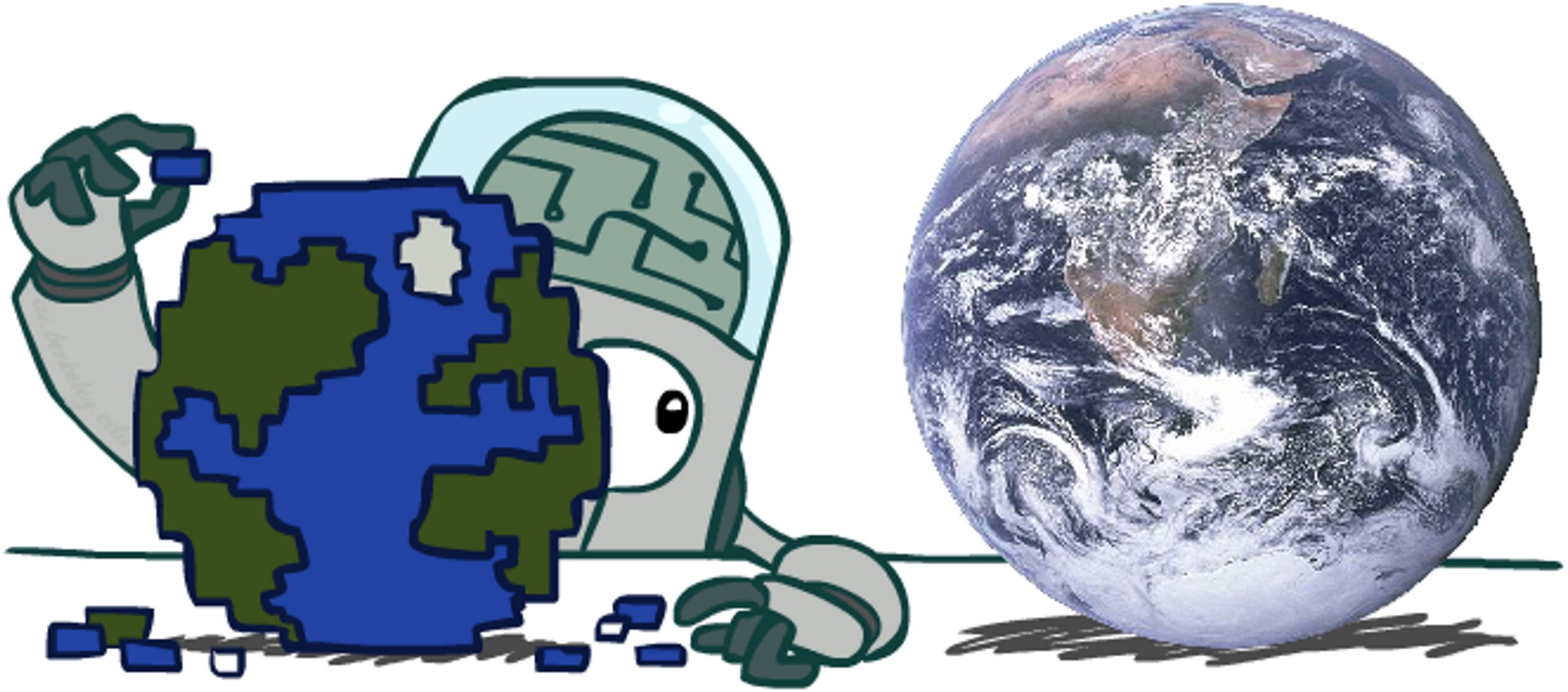
- **state space**
  S
- **actions** in each state
  $a \in A(s)$
- **successor function**
  $\mathrm{s}' = \mathrm{T}(\mathrm{s}, \mathrm{a})$
- action **cost** (reward model)
  $R(s, a, s')$
- **start state**
  $s_0 \in S$
- **goal test**
  $G(s)$

A **solution** to a search problem is a sequence of **actions** that reaches a **goal state**

An *optimal* solution has the least cost among all solutions

Search Problems are *Models*

FOUNDATIONS



State Space

FOUNDATIONS

world state size = $120 (2^{30}) (12^2) 4$

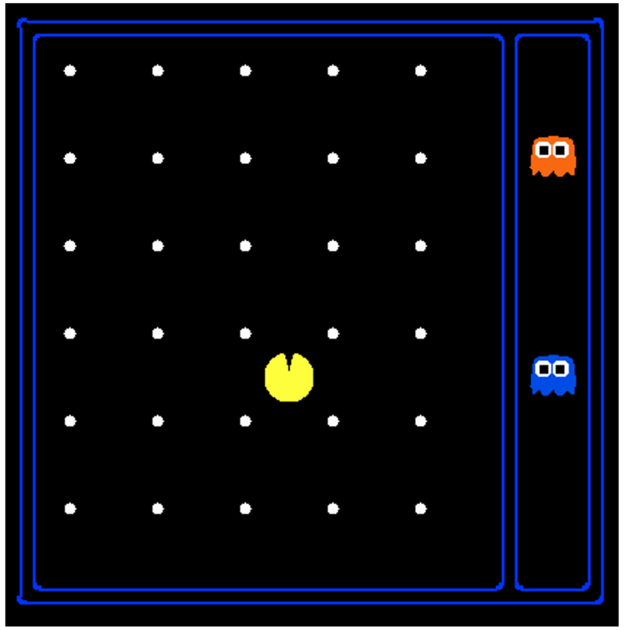
The world state includes every detail of the environment

The search **state space** tracks only the details needed for planning

Pathing

- states: (x, y) location *n = 120*
- actions: NSEW *n = 4*
- successor: location update
- goal test: (x, y) = END

search space size = **120**



Eat All Dots

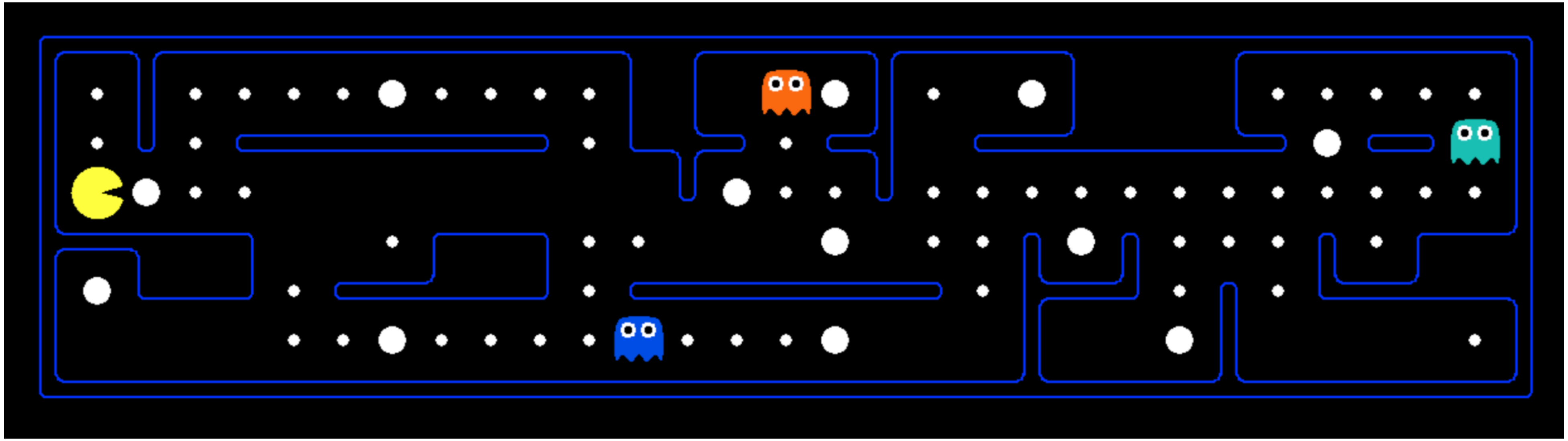
- states: {(x, y), dot booleans} *n = ?*
- actions: NSEW *n = 4*
- successor: location, dot update
- goal test: dots all false

search space size = $120 (2^{30})$

Scared Ghosts

EXAMPLE

Problem: Eat all dots while keeping the ghosts constantly scared



requires knowing Pacman position, dot booleans, power pellet Booleans, remaining scared time

Questions?

live and on sli.do #cse473

that's it for today!

SUMMARY

- *What's hard about modeling the world?*
- Agents and Agent Types
- Rational Behavior
- Environments and State Spaces

UPCOMING

- Uninformed Search Algorithms
- Search with Heuristics

REMINDERS

- Check Ed, Gradescope access
- Complete **Practice Problem 2**
- Do **Project 0** (recommended!)