An Introduction to Reinforcement Learning

CSCI-P556 Applied Machine Learning
Lecture 25

D.S. Williamson

Agenda and Learning Outcomes

Today's Topics

- **Topics**:
  - Intro to Reinforcement Learning
- **Announcements:**
  - **Quiz#3 on Tuesday (4/20)**
    - Support Vector Machines
    - Decision Trees
    - Ensemble Learning
  - **HW#4 Posted Due** (4/21)

3 Learning Formalisms

Learning algorithms can be classified based on the three types of feedback that the learner has access to:

1. **<u>Supervised Learning:</u>** One extreme → For every input, the learner is provided with a target.
   1. The “environment” tells the learner what the target is.
   2. The learner then compares its actual response to the target and adjusts to produce a more appropriate response the next time it receives the same input.

2. **<u>Unsupervised Learning:</u>** On the other extreme → The learner receives no feedback from the world at all.
   1. The learner's task is to re-represent the inputs in a more efficient way, as clusters or using a reduced set of dimensions.
   2. Unsupervised learning is based on the similarities and differences among the input patterns. It does not result directly in differences in overt behavior because its "outputs" are really internal representations.

3 Learning Formalisms

- **How did you learn to ride a bike? Did you use supervised or unsupervised learning?**
  - Neither of the above
  - Trial and error!
  - Falling down hurts!

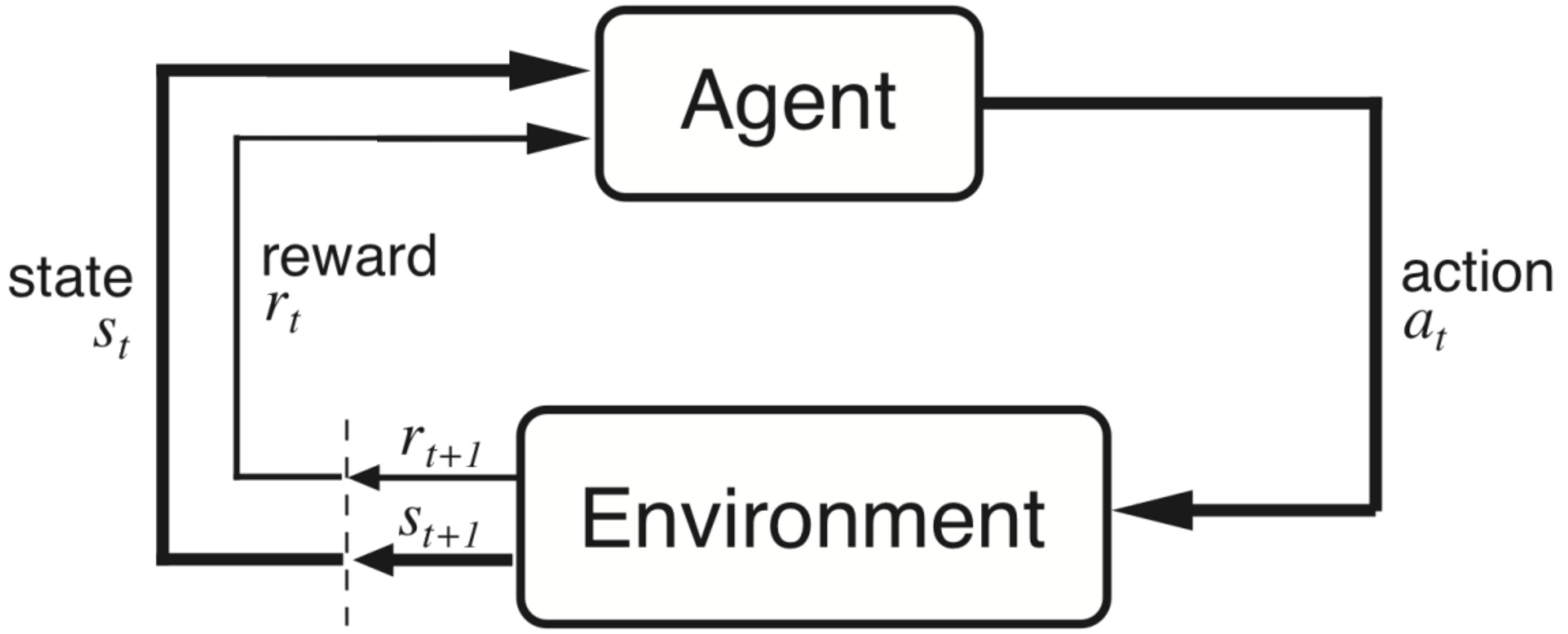
3. Reinforcement Learning: A third alternative, much closer to supervised than unsupervised learning → The learner receives feedback about the appropriateness of its response.

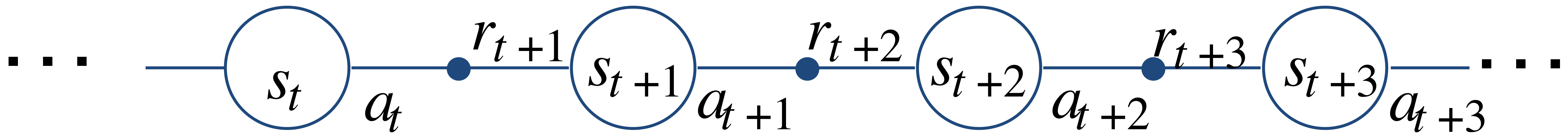
- RL resembles supervised learning, since the learner receives information that what it did is appropriate. However, the two forms of learning differ significantly for errors.
- **Reinforcement Learning only says that the behavior was inappropriate and (usually) how inappropriate it was.**

The Agent-Environment Interface

Reinforcement Learning

- **Agent and Environment interact at discrete time steps**: $t = 0,1,2,\cdots,K$
  - Agent observes **state** at step t: $s_t \in S$
  - Observation produces **action** at step t: $a_t \in A(s_t)$
  - Agent gets resulting **reward** and **state** from environment at the next step: $r_{t+1} \in \mathscr{R}$ and s_{t+1}



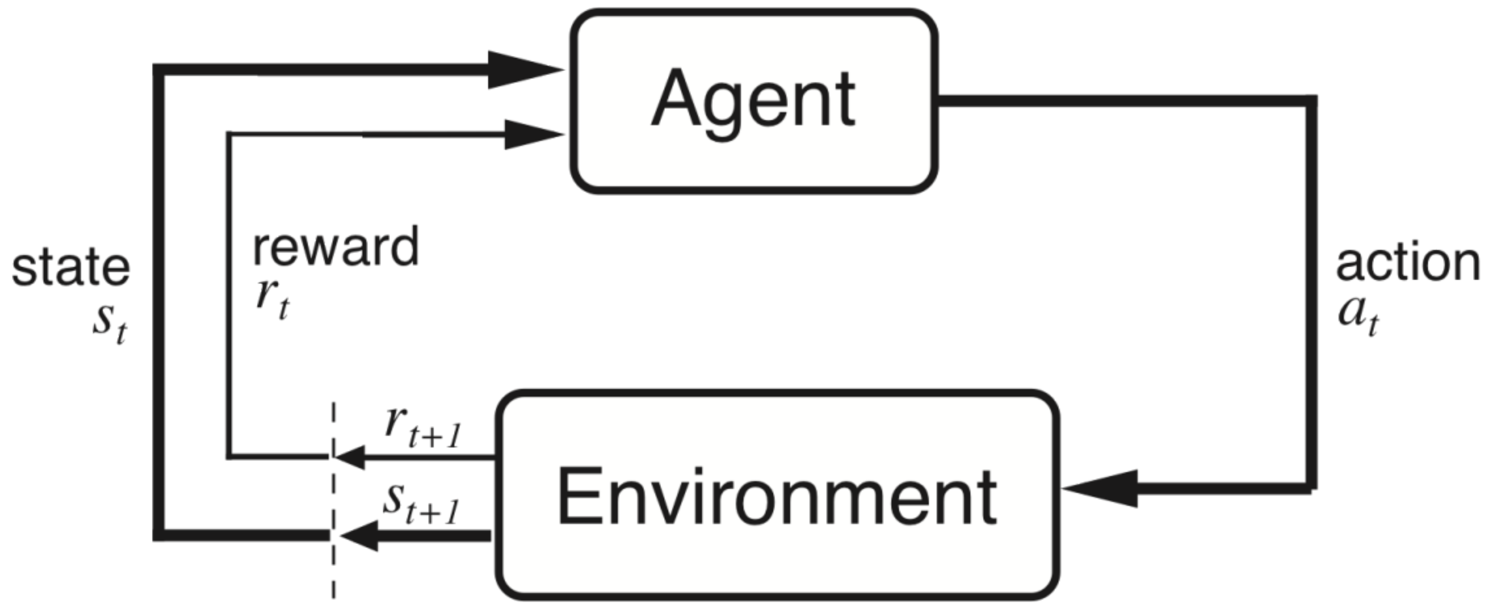


Components of an RL Agent

- The RL agent may include one or more of the following:
  - **Policy**: Defines agents behavior (selected action) in a provided state
  - **Value function**: Agent’s evaluation of the state (e.g. how good is it?) and/or action
  - **Model**: Agent’s representation of the environment (e.g. how the environment thinks/responses)

What does the agent learn?

A Policy to Select Best Actions!



- Straight line **deterministic policy:** $a = \pi(s)$
  - No guarantee that agent will reach goal!
- Alternatively, the actions can have probabilistic effects and depend on the state (or observation).

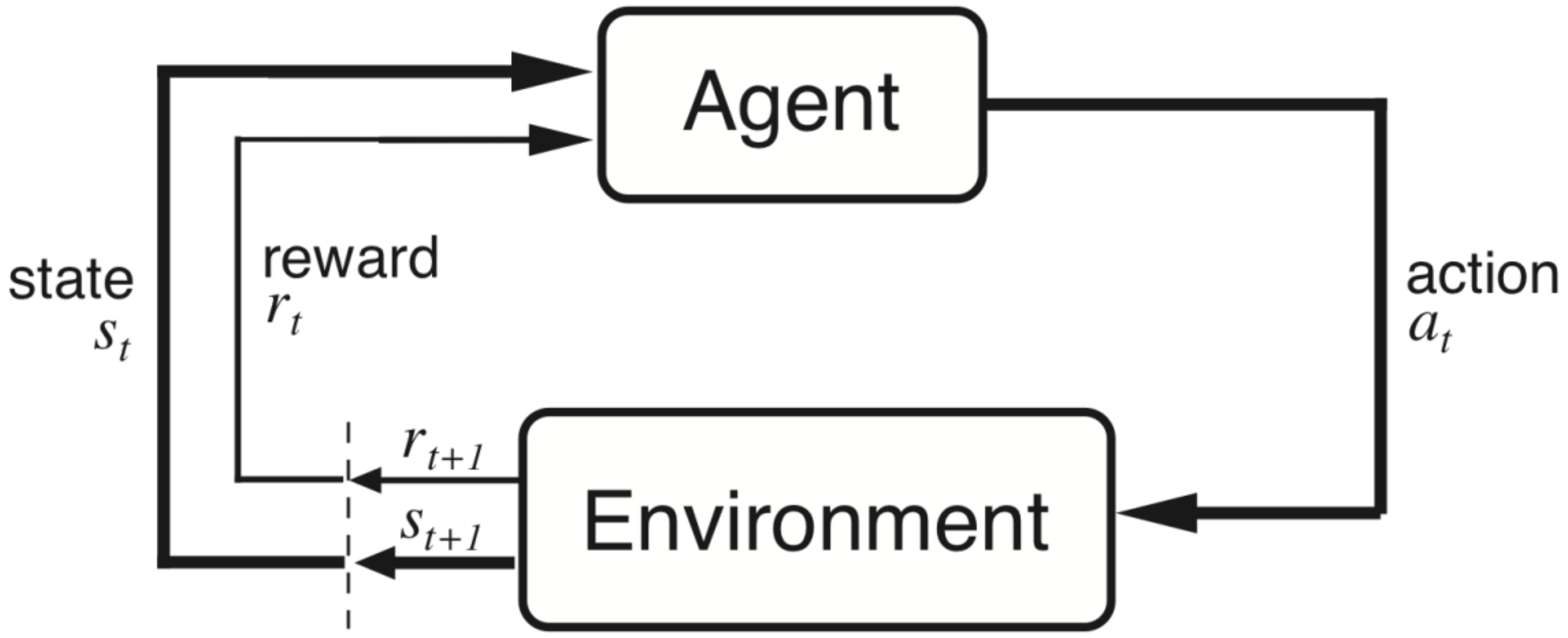
Stochastic Policy: $\pi(a\,|\,s) = P(A_t = a\,|\,S_t = s)$

Map from states to actions.

Agent tries to learn the optimal policy. But optimal in terms of what?

Rewards

From Environment to Agent, based on Action/State

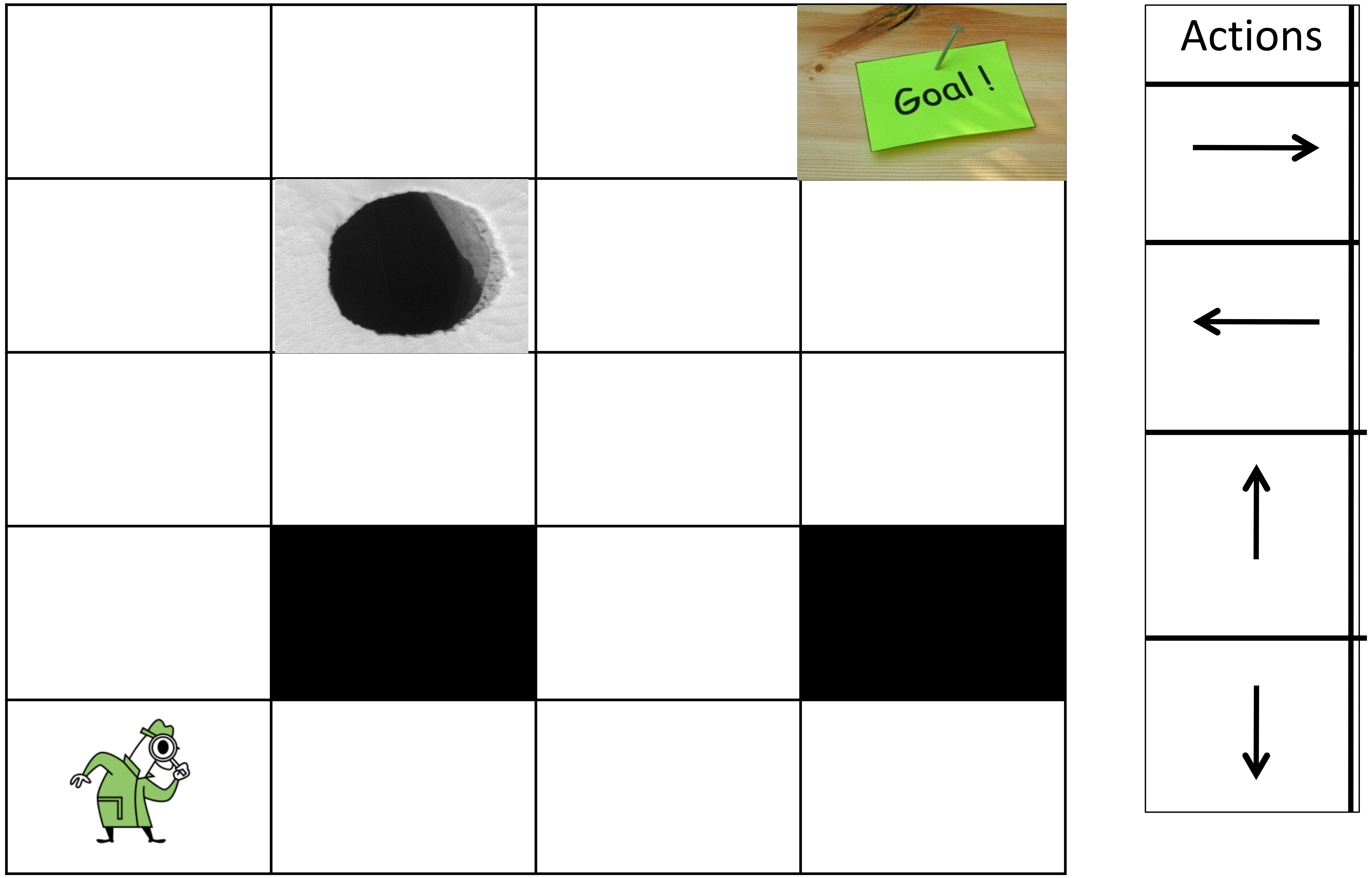


- Suppose the sequence of rewards **after** step t is: $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$
- **Total future return over all time**

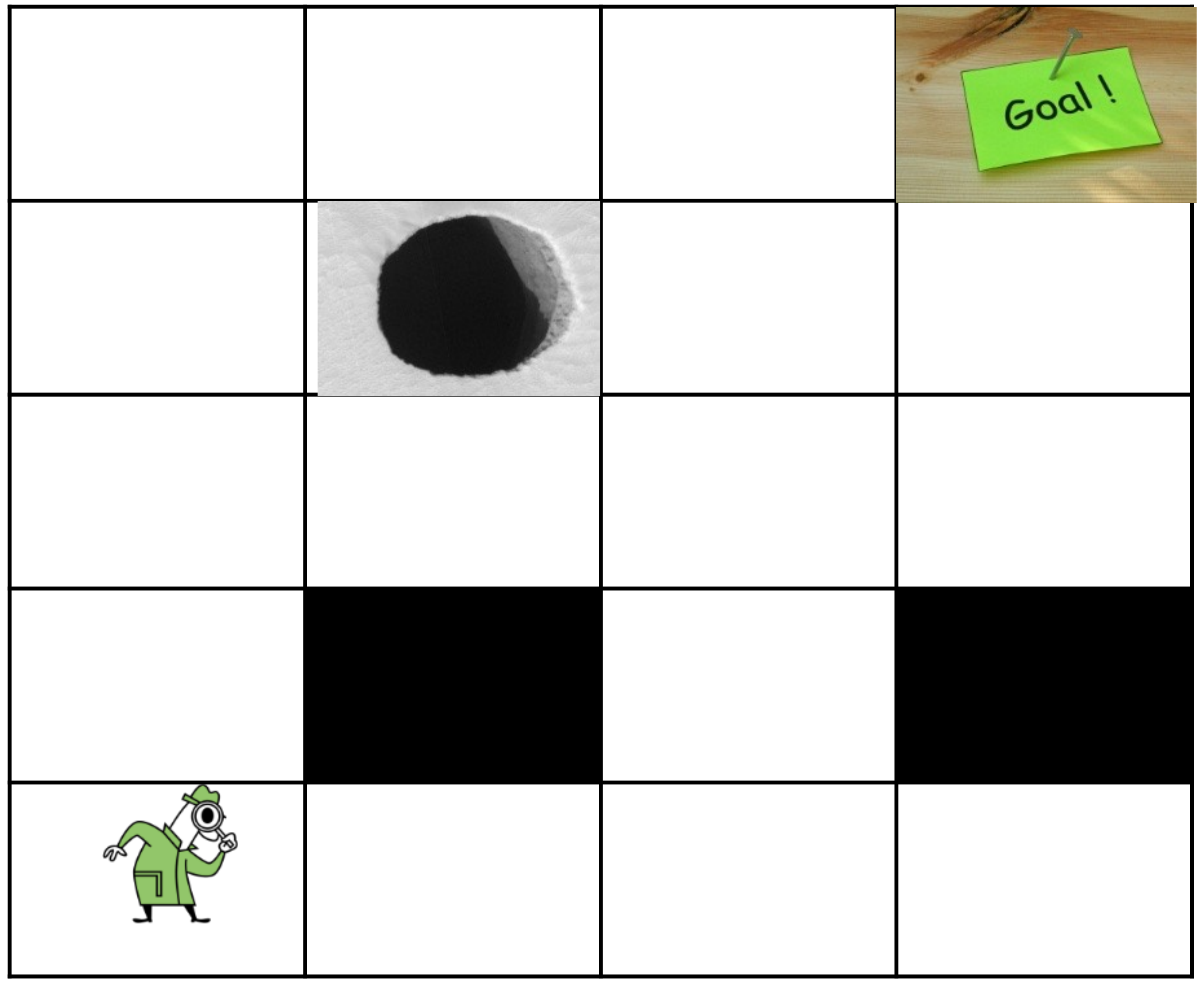
$$R_t = r_t + r_{t+1} + r_{t+2} + \ldots + r_\infty,$$

RL Example

Manuever Man to Reach Goal



Example domain - Rewards



Example – Reward 1

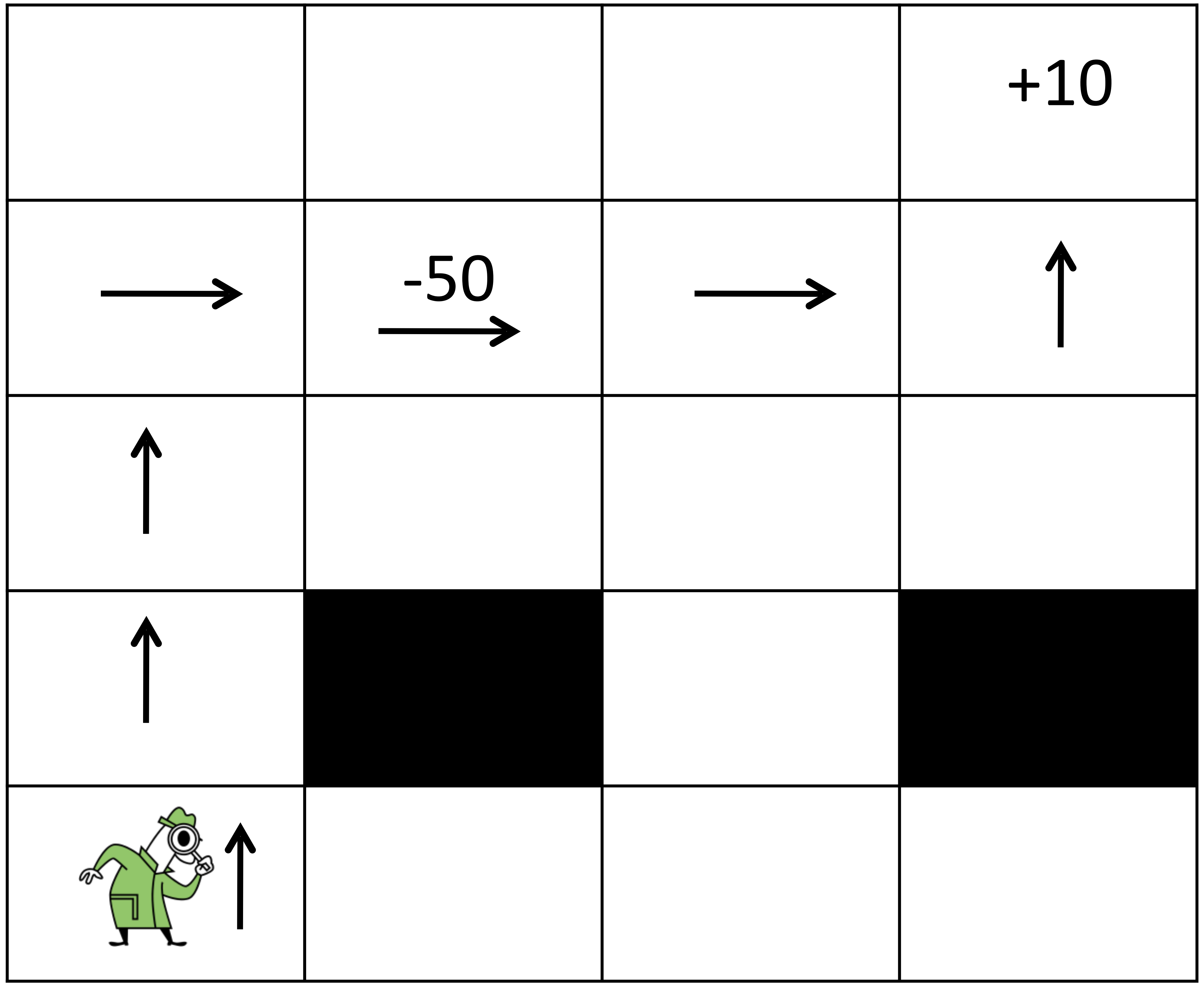
0 for every step taken
+10 for reaching the goal
-50 for falling in the pit

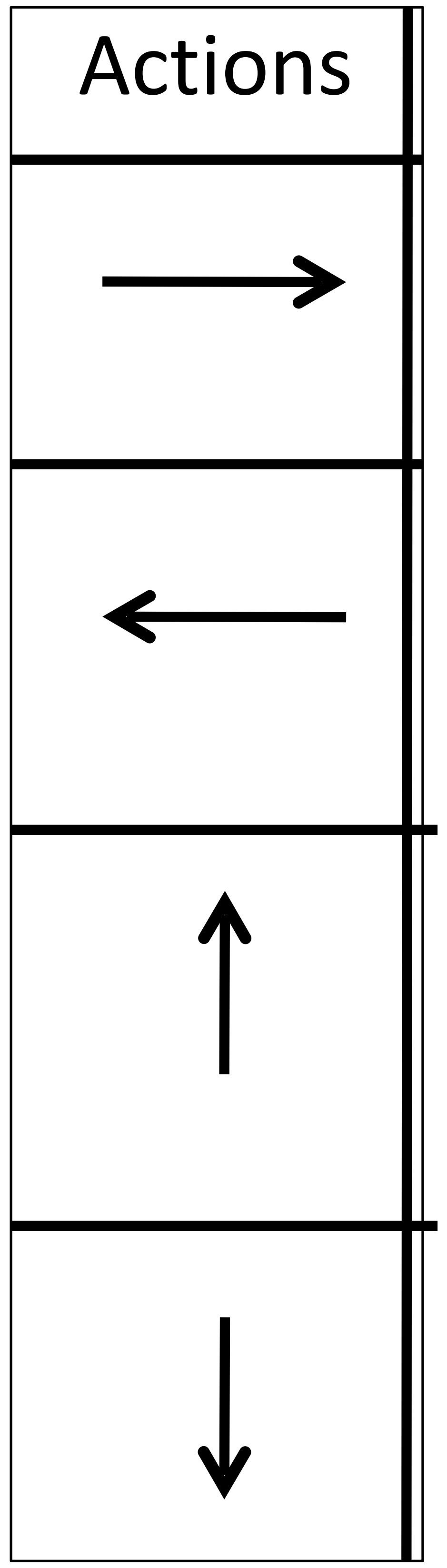
Example – Reward 2

-1 for every step taken
+10 for reaching the goal
-50 for falling in the pit

What is the difference in optimal policy between Reward 1 scheme and Reward 2 scheme?

Example domain





Example – Reward 1

0 for every step taken
+10 for reaching the goal
-50 for falling in the pit

Reward Sequence: 0 + 0 + 0 - 50 + 0 + 0 + 10 = -40

Rewards

From Environment to Agent, based on Action/State

Problem: Future returns are all equally important!

$$R_t = r_t + r_{t+1} + r_{t+2} + \ldots + r_\infty,$$

Solution: Discounted reward - Prioritize current reward ($\gamma \in [0,1]$)

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$$

Optimality Criterion: Find π that maximizes $E[R_t]$

Note that the goal has changed from maximizing reward to expected reward

Value Functions

The agent "values" certain states more than others

- The **value of a state** is the expected return starting from that state; depends on the agent's policy:
- It is modeled as a function of the state
- It is a prediction of the expected total future reward. (e.g. used by the agent to choose between actions)

$$v_\pi(s) = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid S_t = s]$$

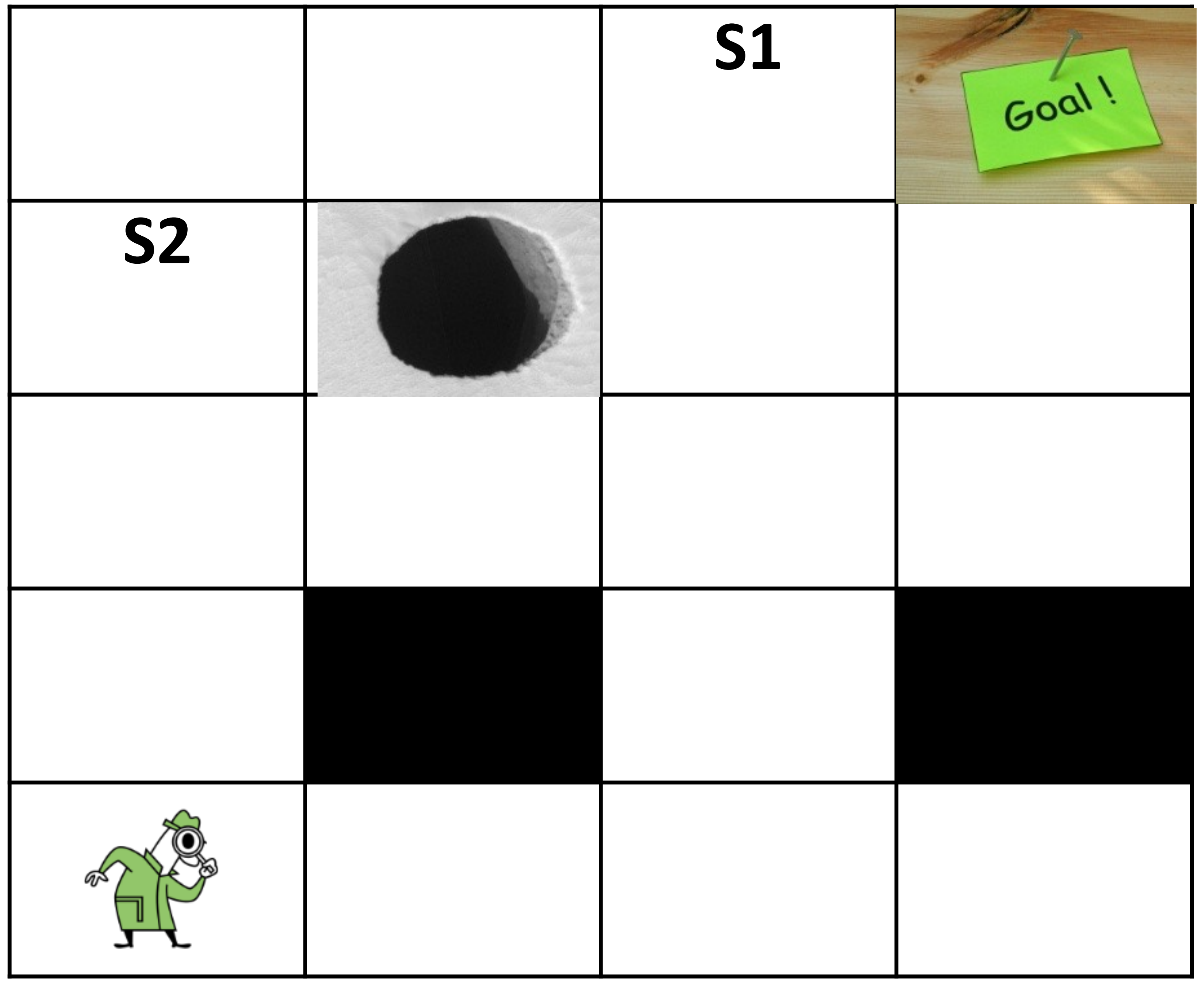
Example domain – Values

Which of the states will have a higher value? S1 or S2?

$$v_\pi(s) = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid S_t = s]$$

Which action will have a higher value in S1? → or ↓ ?

Which action will have a higher value in S2? → or ↑ ?



General RL Processing

- Given a policy:
  - Evaluate the **policy** using the **value** function.

$$v_\pi(s) = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid S_t = s]$$

  - **Improve** the policy **by acting greedily** with respect to the value function

$$\pi(s) = \arg\max_\pi F(v_\pi)$$

Model of the Environment

An optional component

- **Used by the agent to predict what the environment will do next**
  - $\mathscr{P}$ predicts the next state
  - $\mathscr{R}$ predicts the next (immediate) reward

$$\mathscr{P}^a_{ss'} = P(S_{t+1} = s' \,|\, S_t = s, A_t = a)$$

State Transition Probability

$$\mathscr{R}^a_s = E(R_{t+1} \,|\, S_t = s, A_t = a)$$

Expected Reward

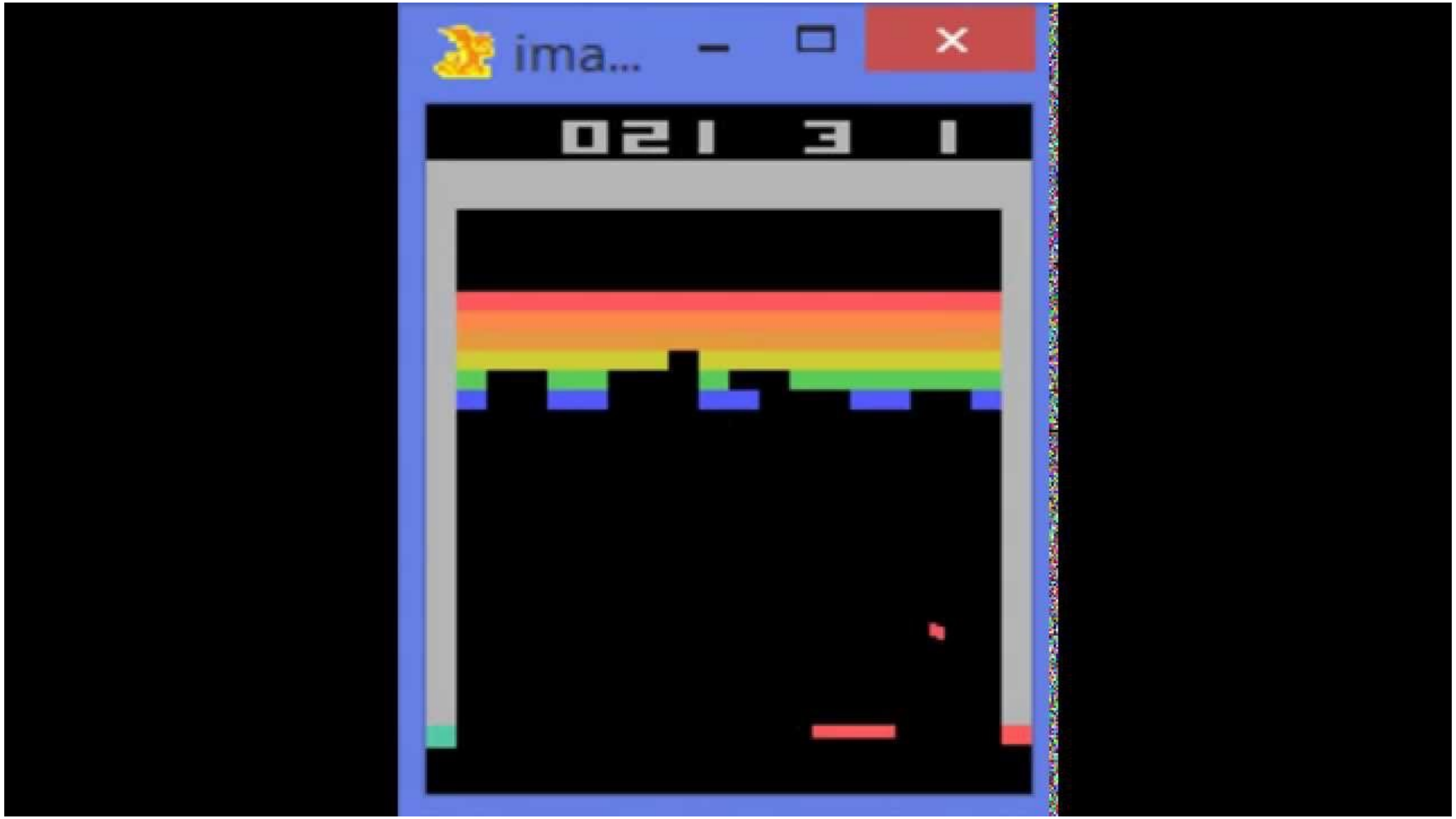
Categories of RL agents

- **Value Based**
  - ~~No policy~~
  - Value function
- **Policy Based**
  - Has Policy
  - ~~No Value Function~~
- **Actor-Critic:**
  - Policy
  - Value Function

- **Model Free**
  - Policy and/or Value Function
  - ~~No Model~~
- **Model Based**
  - Policy and/or Value Function
  - Model

RL Example

https://youtu.be/V1eYniJ0Rnk



Other Resources

- **At IU:**
  - Dr. Roni Khardon's course on RL
    - B659: Topics in AI: Learning Planning and Acting in Complex Environments (Short title: Reinforcement Learning)
- **External**:
  - Sutton and Barto's, An Introduction to Reinforcement Learning
  - David Silver's Tutorial

Next Class:

Quiz #3