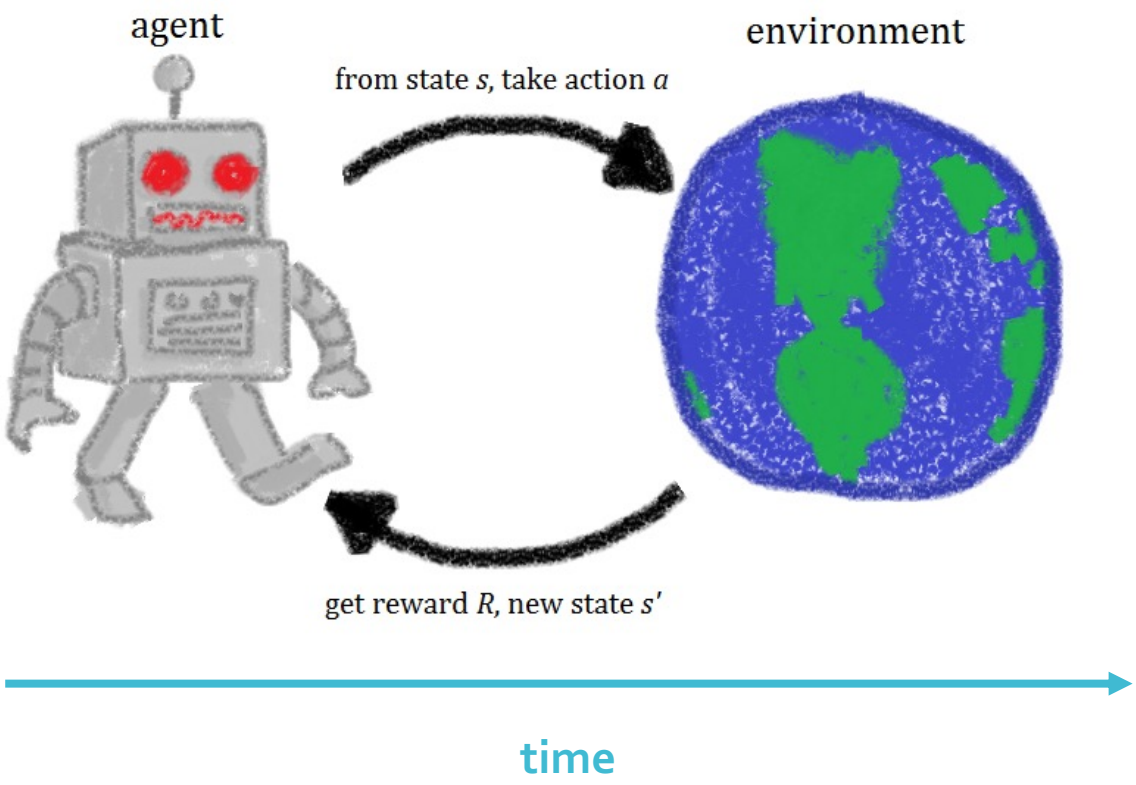
Basic RL.1

Judy Tutorial

What is RL?


agent
environment
from state s, take action a
get reward R, new state s'
time

What are env and agent?

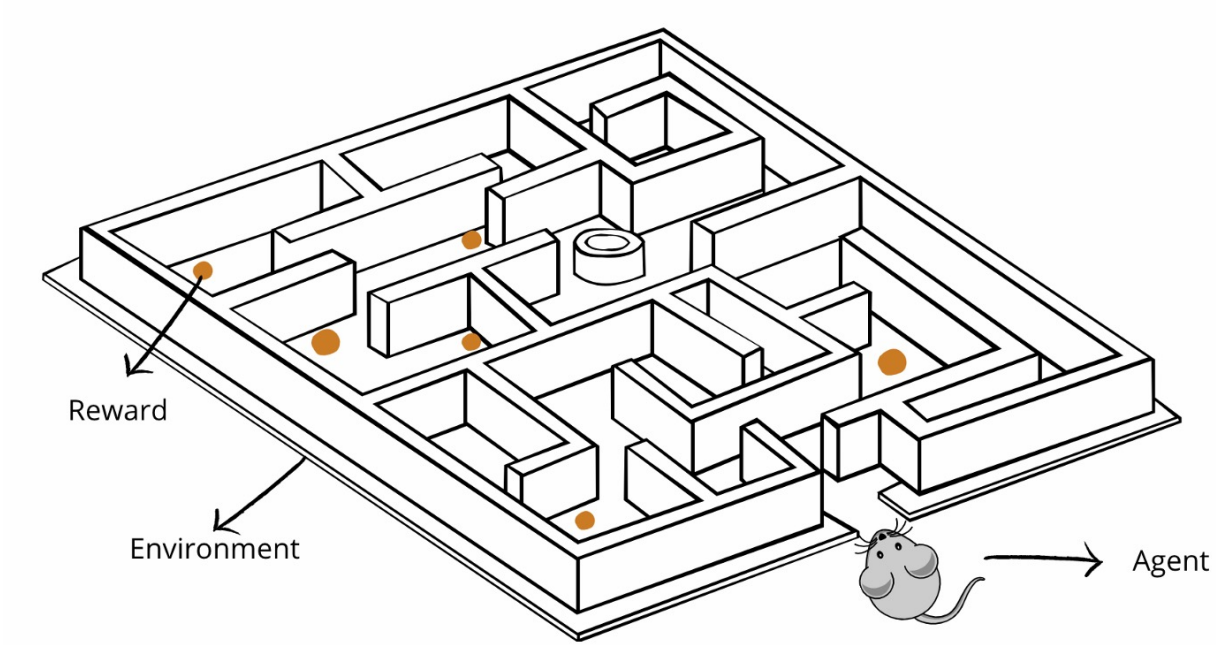
Env

- **The world!** where **the Agent** lives and interacts, provides **states** and **rewards**

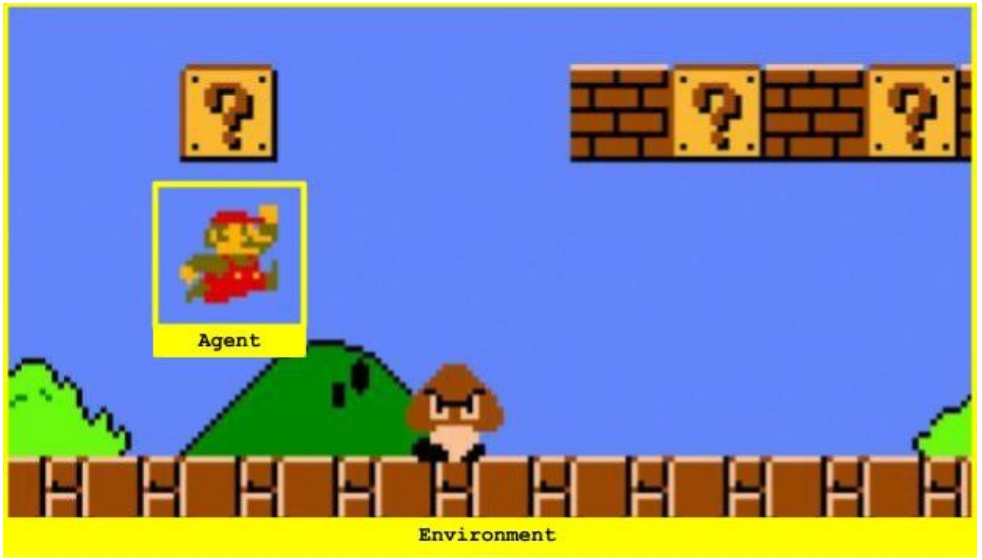
Agent

- can perform some **actions** to **the Env**, but cannot influence the **dynamics** of **the Env**

Mouse <-> Maze
(agt) (env)



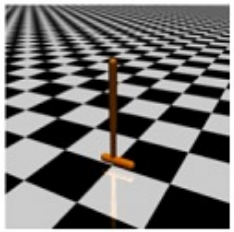
SuperMario (agt) <-> Mushroom World (env)



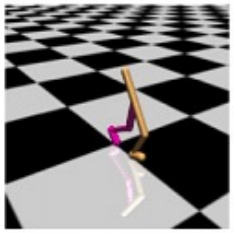
self-driving car (agt) <-> road env (env)

robot (agt) <-> physical engine / real world (env)


Swimmer


Hopper

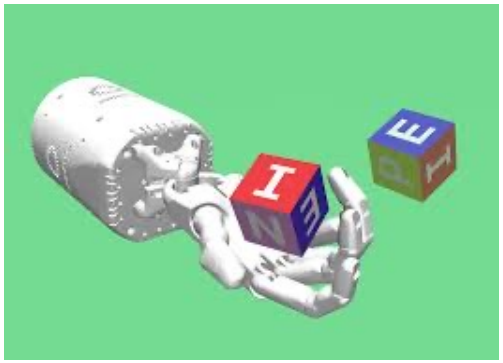

Half Cheetah

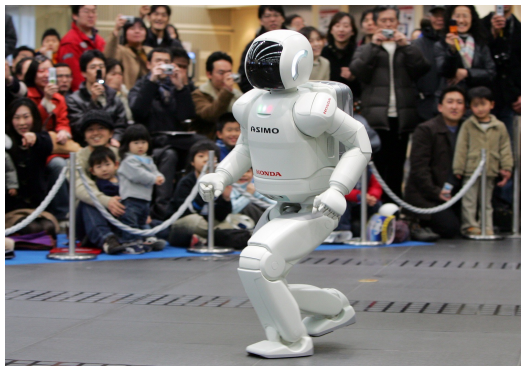

Walker


Ant

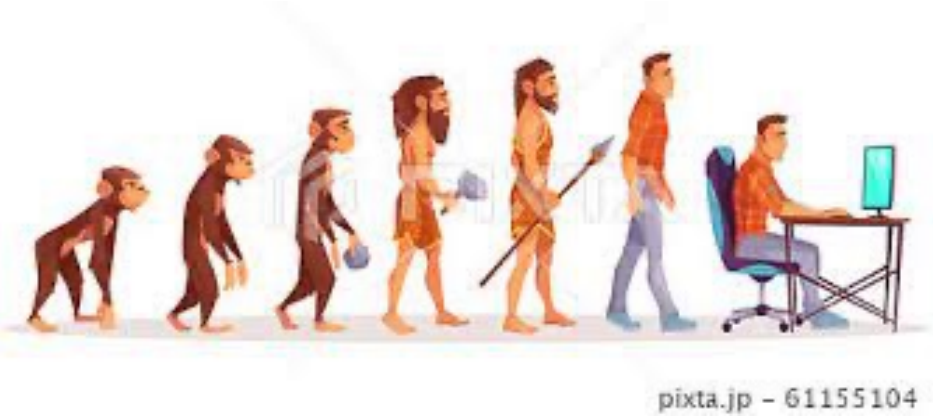

Simplified Humanoid


Full Humanoid

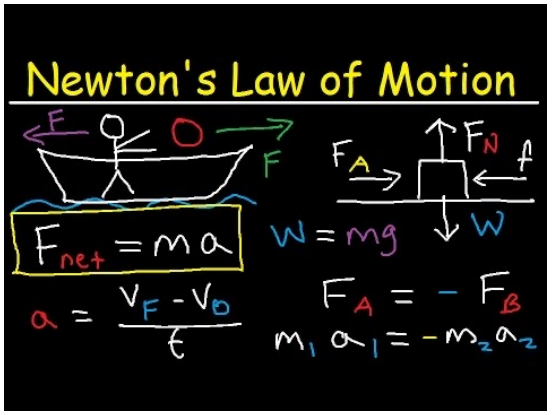




Human <-> Earth
(agt) (env)



the laws of newton physics



What are these?

States: $s \in \mathcal{S}$

Actions: $a \in \mathcal{A}$

Policy: $\pi(a|s) \in [0, 1]$

Rewards: $r(s, a)$ numerical feedback

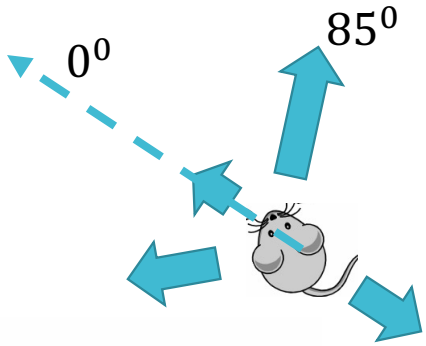
Dynamics: $p(s'|s, a) \in [0, 1]$

Next state

Markov Decision Process

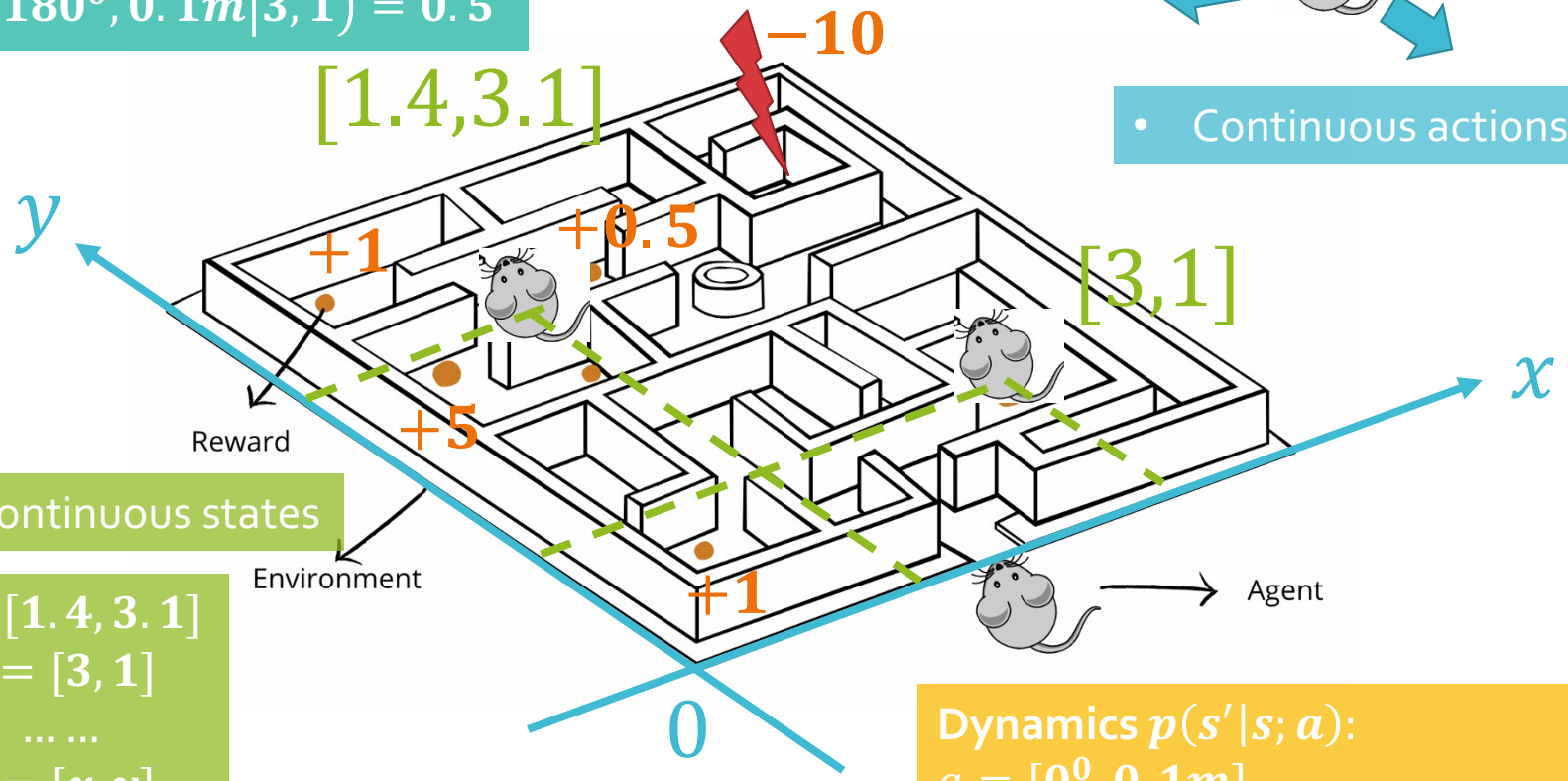
Mouse (agt) <-> Maze (env)

$a = [\varphi, d]\ or\ +[v, a]$



Policy $\pi(a|s)$:
$a = [0^0, 0.1m]$
$s = [3,1]$
$\pi(0^0, 0.1m|3,1) = 0.5$
$\pi(180^0, 0.1m|3,1) = 0.5$

- Continuous actions



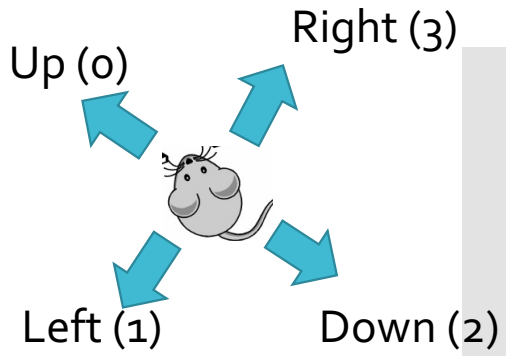
- Continuous states

$s = [1.4, 3.1]$
$s = [3, 1]$
… …
$s = [x, y]$

Dynamics $p(s'|s; a)$:
$a = [0^0, 0.1m]$
$s = [3,1]$
$p(3, 1.1|3, 1; 0^0, 0.1cm) = 0.9$

Mouse (agt) <-> Maze <simplified> (env)

10×8 maze



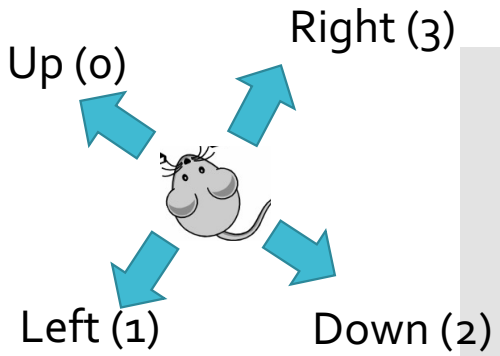
- Discrete states

$s = [0, 0]$
$s = [0, 1]$
… …
$s = [0, 7]$

- Discrete actions

Mouse (agt) <-> Maze <simplified> (env)

10×8 maze



- Discrete actions

Stochastic Policy $\pi(a|s)$:

$a = [0]$

$s = [4,7]$ start point

up $\sim \pi(0|4,7) = 0.25$

Left $\sim \pi(1|4,7) = 0.25$

right $\sim \pi(3|4,7) = 0.25$

Down $\sim \pi(2|4,7) = 0.25$

Deterministic Policy $\pi(s)$:

up $\sim \pi(0|4,7) = 1$

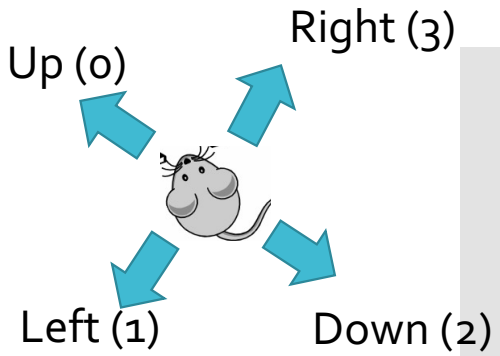
Left $\sim \pi(1|4,7) = 0$

right $\sim \pi(3|4,7) = 0$

Down $\sim \pi(2|4,7) = 0$

Mouse (agt) <-> Maze <simplified> (env)

10×8 maze

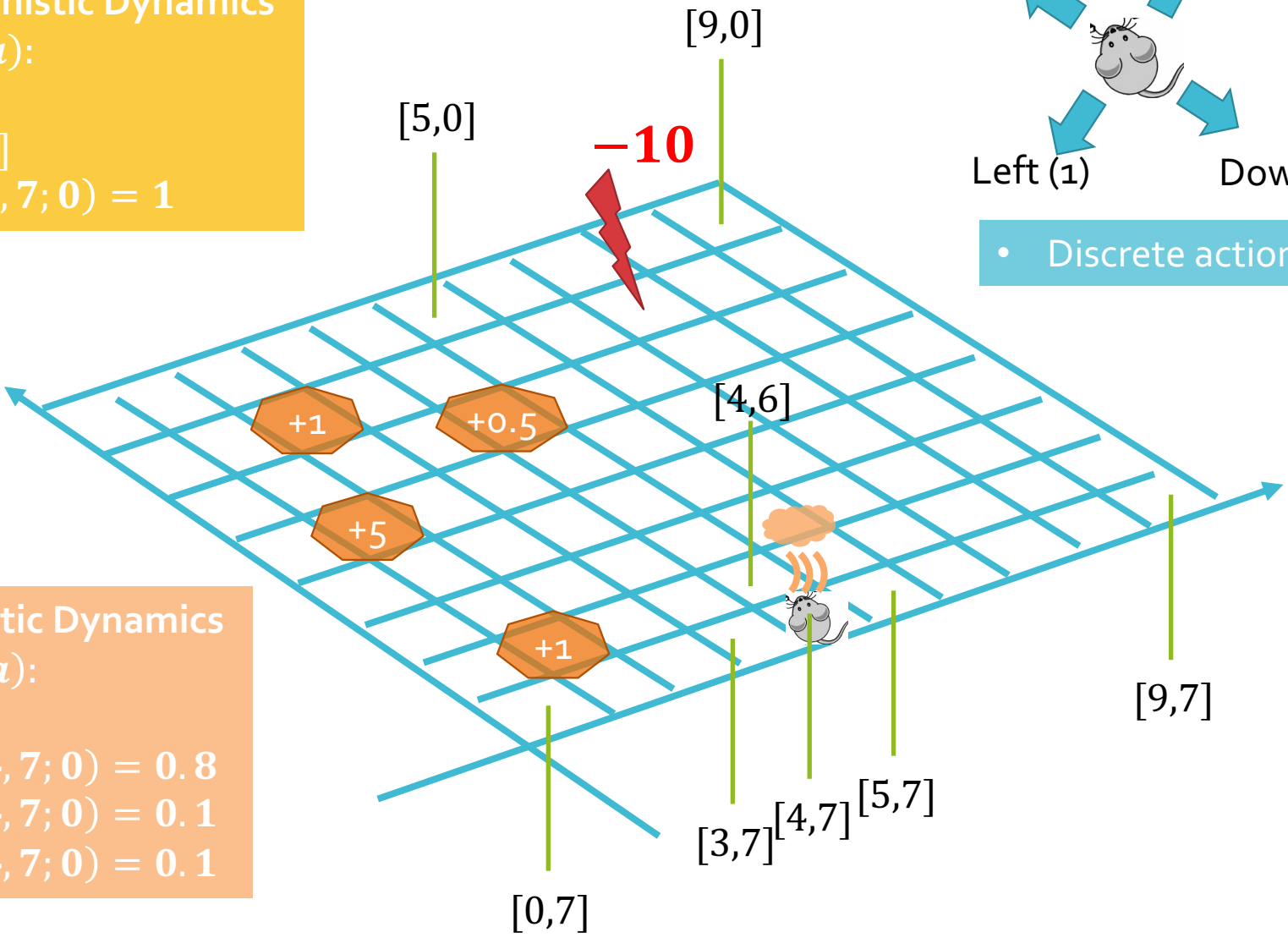


- Discrete actions

Deterministic Dynamics
$p(s'|s;a)$:
$a = [0]$
$s = [4,7]$
$p(4,6|4,7;0) = 1$

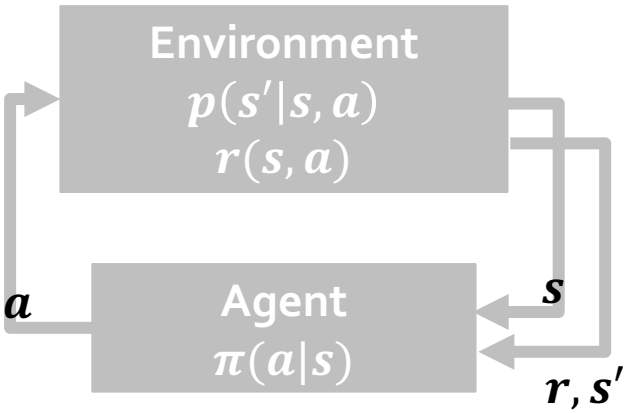
Stochastic Dynamics
$p(s'|s;a)$:

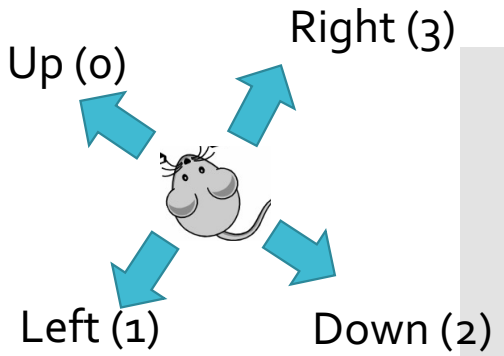
$p(4,6|4,7;0) = 0.8$
$p(5,7|4,7;0) = 0.1$
$p(3,7|4,7;0) = 0.1$



Mouse (agt) <-> Maze <simplified> (env)

10×8 maze





- Discrete actions

"MDP is **abstract** and **flexible** and can be applied to many different problems in many different ways."

-- Richard Sutton

time step

Any!

- neither restricted to real time nor to fixed intervals
- refer to arbitrary successive states of decision making and acting

states

Low-level:

- Sensations, *such as direct sensor readings*

High-level:

- Symbolic descriptions *of objects in a room*
- Even can be entirely *mental* or *subjective*

i.e. in the state of not being sure, or being surprised

Can be anything we can know that might be useful in making a decision

states

- Scalar

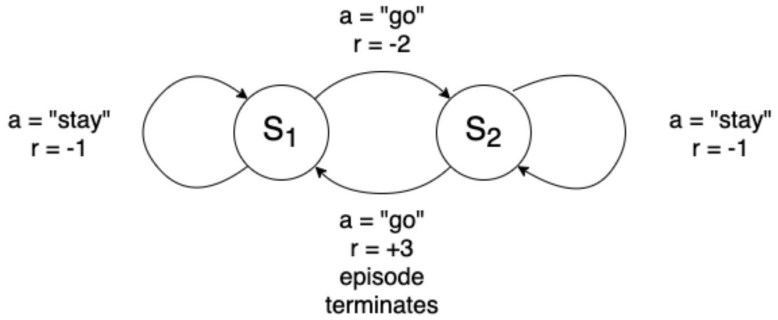
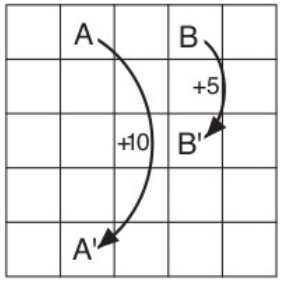


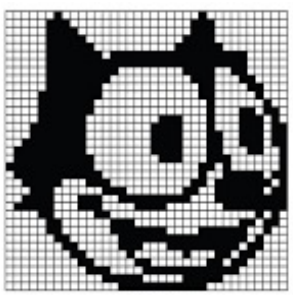
Figure 1: 2-state MDP

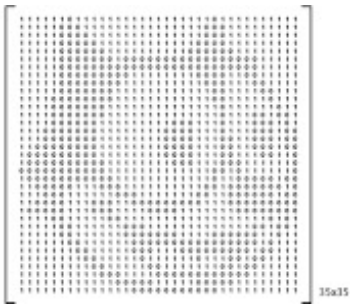
- Vector

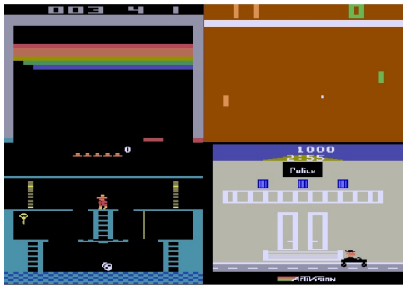


Actions

- Matrix







actions

Low-level:

- *Up, down, left, right*
- *voltages applied to the motors of a robot arm*

High-level:

- Whether or not to have lunch or go to graduate school?
- Can be *mental* or *computational*

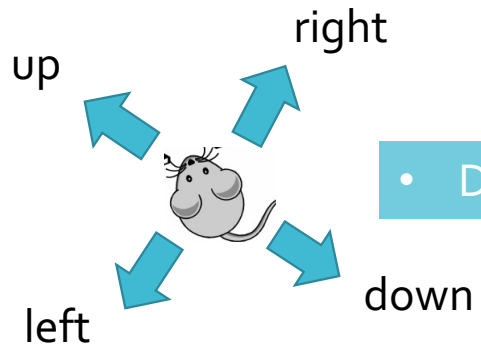
i.e. Some actions might control what an agent chooses to think about

or where an agent should focus its attention

Can be any decisions we want to learn how to make

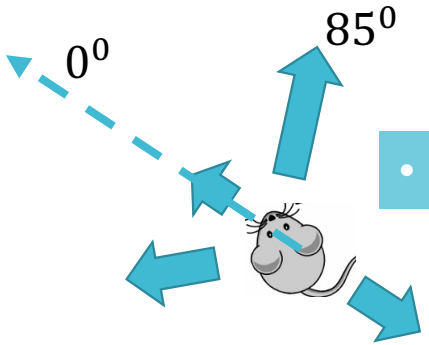
actions

- Scalar $0,1,2,3$



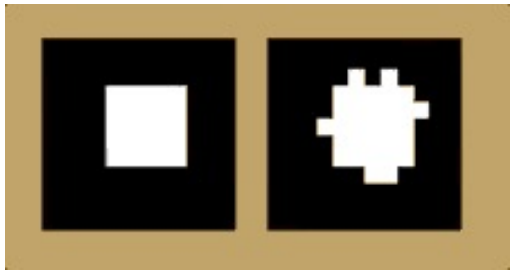
- Discrete actions

- Vector $[\varphi, d, v, a]$



- Continuous actions

- Matrix



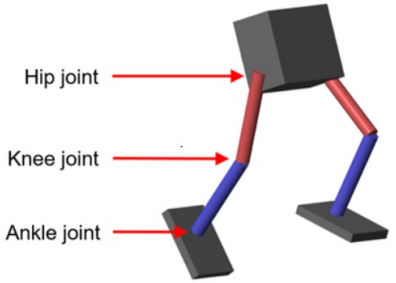
https://ai.stackexchange.com/questions/17225/reinforcement-learning-possible-with-big-action-space

https://www.mathworks.com/help/reinforcement-learning/ug/train-biped-robot-to-walk-using-reinforcement-learning-agents.html#:~:text=This%20reward%20function%20encourages%20the,Tf%20)%20at%20every%20time%20step.

rewards

- robot learn how to walk:

reward on each time step proportional to the robot's forward motion



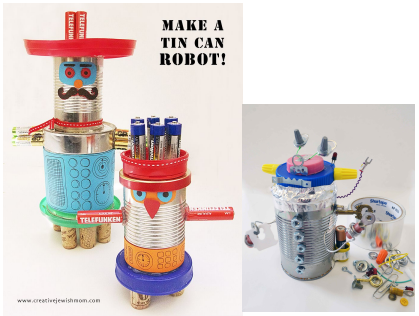
- agent escapes from a maze:

-1 reward on each time step, encouraging the speed

- robot learning to find and collect empty soda cans for recycling:

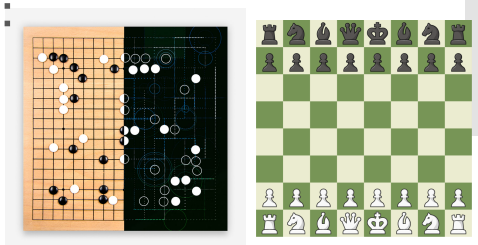
+1 for each can collected, +0 otherwise

-1 if bumping into things or when somebody yells at it



- agent learns to play checkers or chess:

+1 for winning, -1 for losing, 0 for drawing



- Maximizing rewards is **always** aligned with achieving the Goal

- "It is thus critical that the rewards we set up truly indicate what we want accomplished"

- "Reward is your way of communicating to the agent ***what*** you want it to achieve, not ~~***how***~~ you want it achieved"

Chess

Only set reward (+1) for winning

Not for achieving subgoals such as

- Taking its opponent's pieces
- Gaining control of the center of the board

The representation of **states**, **actions** and **rewards** remains more ***art*** than **science**.

Questions

- Can you create more agent and env?
- What are the state, action, reward in SuperMario, self-driving car or robot control?
- Based on the insights we gained so far, what can be inferred for human <-> earth interactions?

reading

- Example 3.1 – bioreactor
- Example 3.2 – pick-and-place robot
- Example 3.3 – recycling robot

"Reinforcement Learning an introduction 2nd edition"

-- Richard Sutton and Andrew Barto