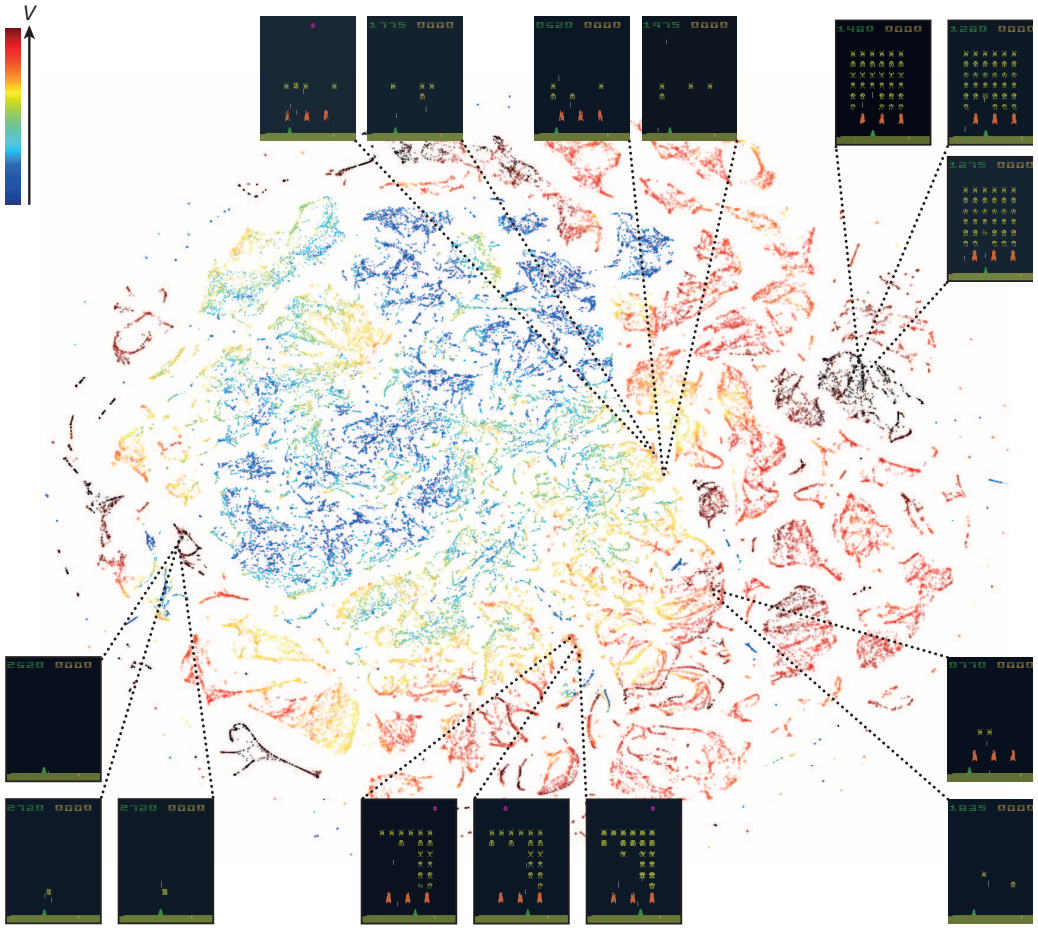
HSLU Lucerne University of Applied Sciences and Arts

Reinforcement Learning Introduction



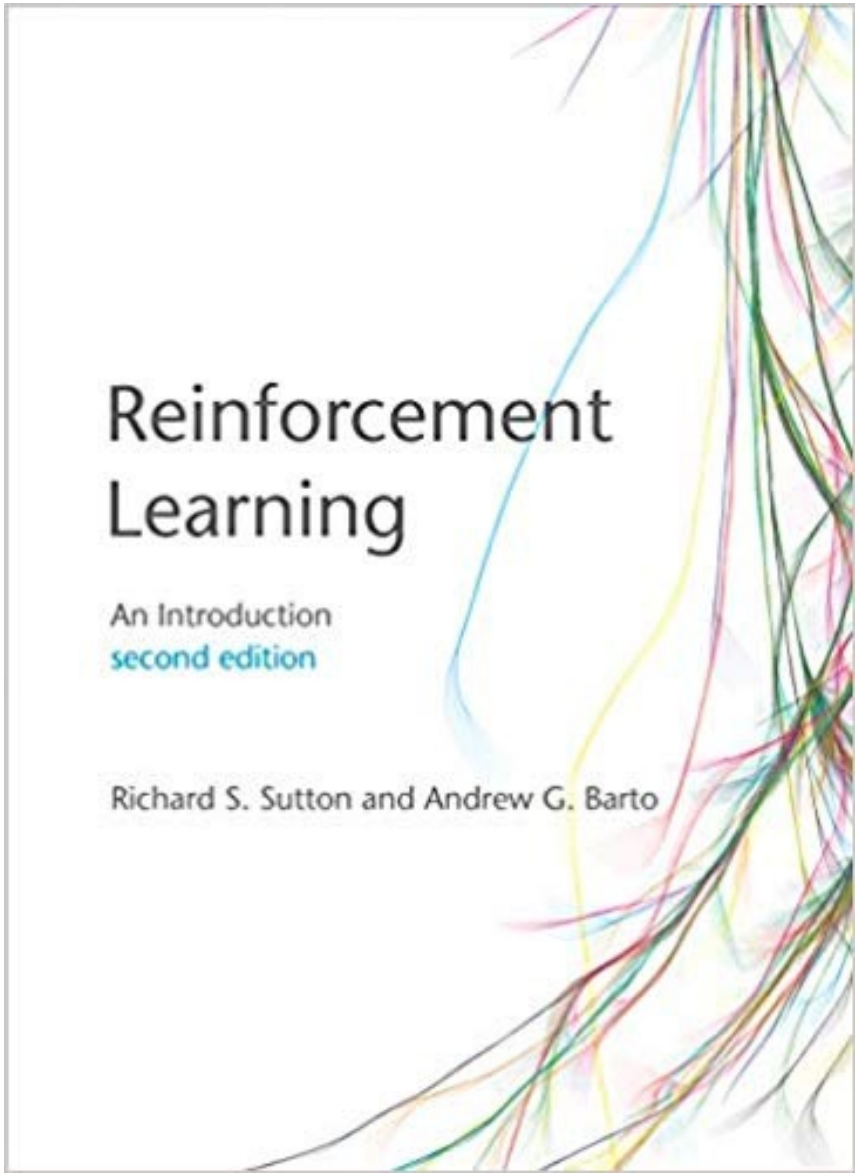
Reinforcement Learning
September 22, 2022

FH Zentralschweiz

Welcome to the course

The course uses the textbook:

Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, 2018, 2nd edition



http://incompleteideas.net/book/the-book.html

Reading assignments are given for each topic

Course Administration

Most topics will contain

- quizzes on Illias and
- exercises.

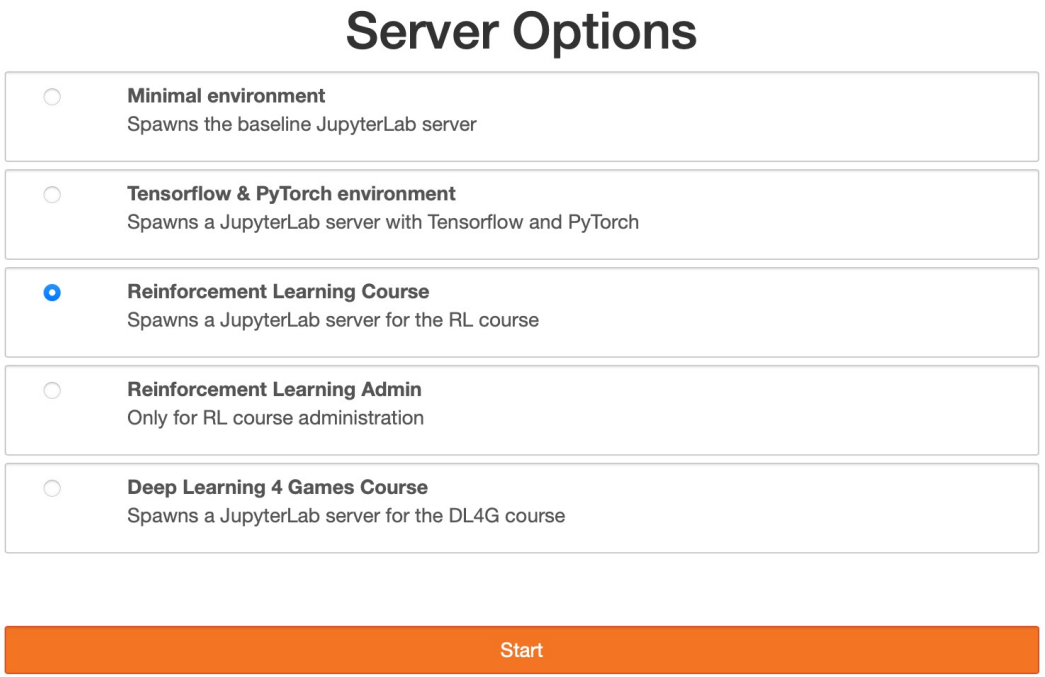
Both are mandatory for the testat. Exercises will give 10 points each, and a minimum number of points is needed for each exercise.

There (should) be enough time during the course to solve the exercises. Exercises are available on a kubernetes cluster using jupyter lab and nbgrader.

Exercise Environement

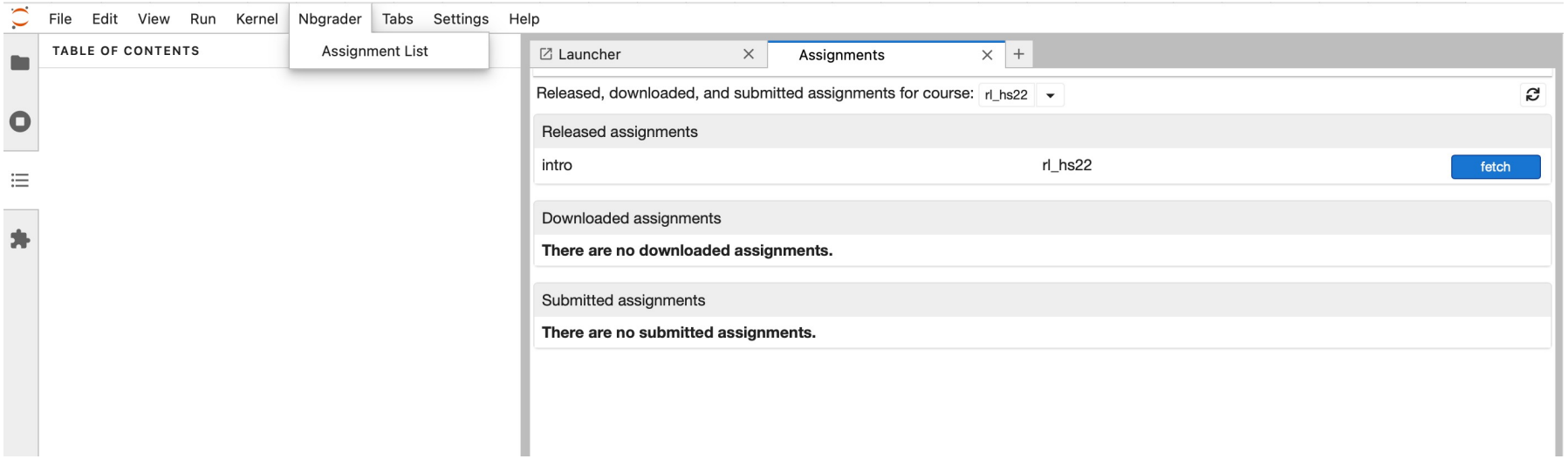
Login to https://gpuhub.el.eee.intern using your enterpriselab account.

Select *Reinforcement Learning Course* Image



Exercises Environment

Select nbgrader->Assignment List



Select *fetch* to get the assignement and *submit* to submit it when solved

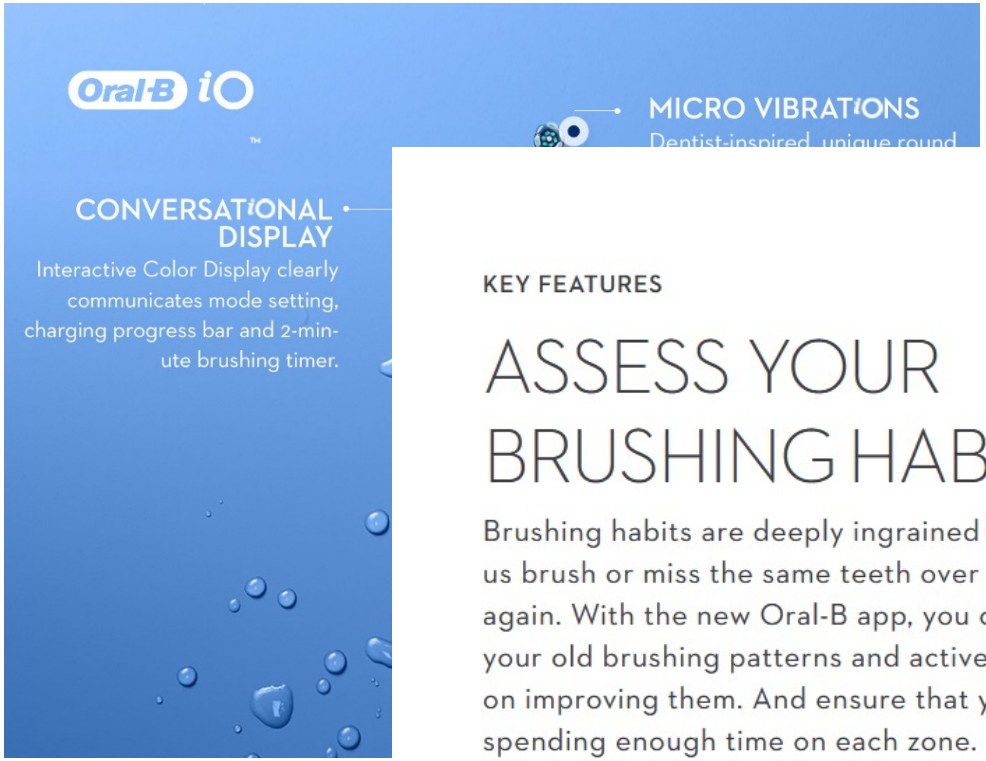
Intro to python

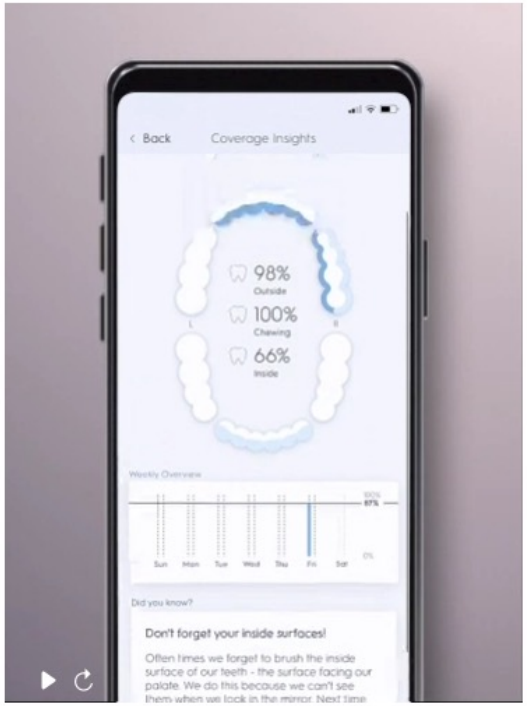
- The exercise are in python 3.
- In the first week there are no exercises, but a python course and some python basic exercises (that do not need to hand in)
- If you have not programmed python yet, please familiarize yourself with python in the first week ☺.

Learning Objectives: Introduction

- Differentiate between Reinforcement Learning (RL) and other Machine Learning (ML) Techniques
- Know when RL methods can be applied and when not
- Explain the interaction of a RL technique with the environment
- Know the different types of RL agents

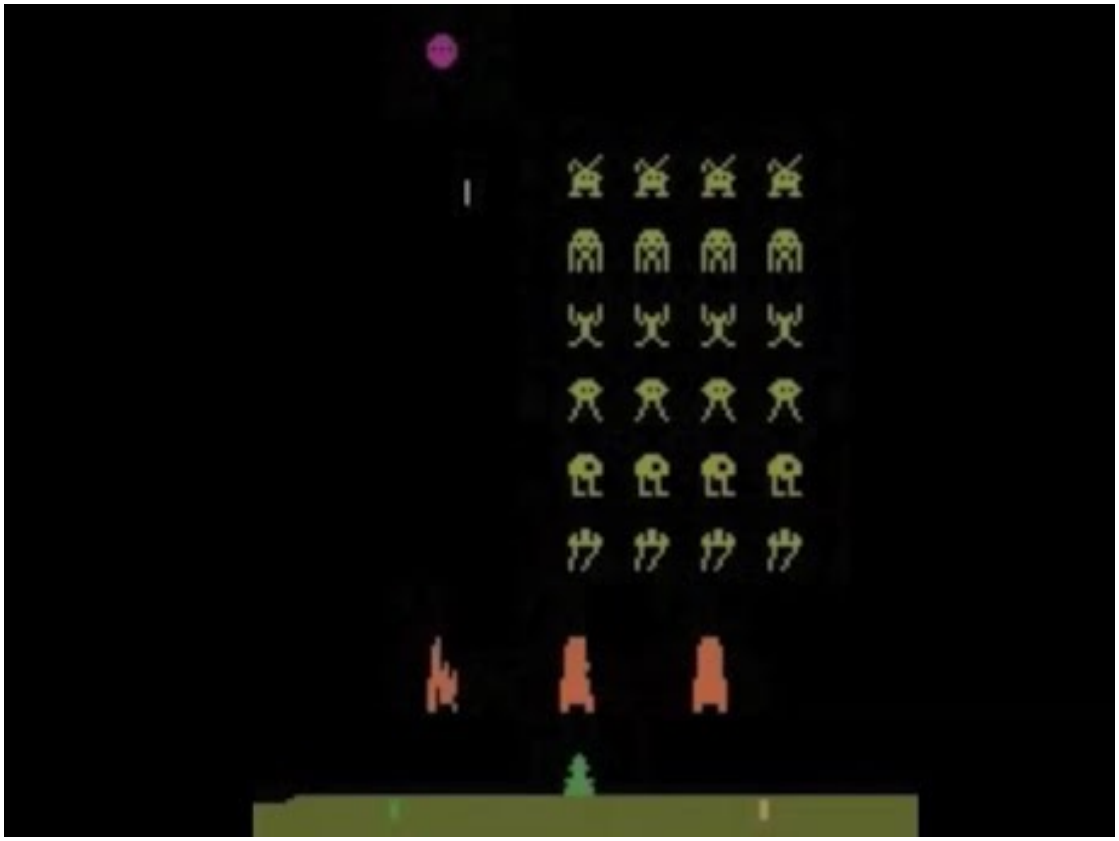
In products...


Oral-B iO
MICRO VIBRATIONS
CONVERSATIONAL DISPLAY
Interactive Color Display clearly communicates mode setting, charging progress bar and 2-minute brushing timer.
KEY FEATURES
ASSESS YOUR BRUSHING HABITS
Brushing habits are deeply ingrained – most of us brush or miss the same teeth over and over again. With the new Oral-B app, you can see your old brushing patterns and actively work on improving them. And ensure that you're spending enough time on each zone.

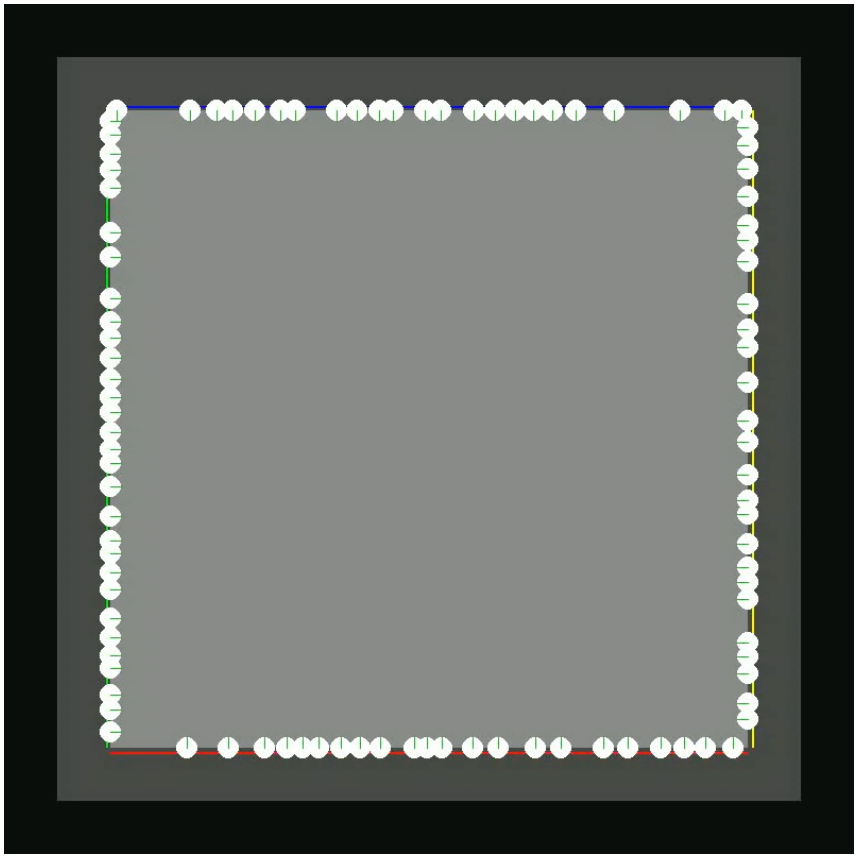

Back
Coverage Insights
98%
100%
66%
Don't forget your inside surfaces!

Example:

Example: Atari Games (Deepmind)



Example: Crossing (HSLU, ABIZ)



Example: Hide and Seek

Reinforcement Learning

What is reinforcement learning?

What is reinforcement learning not?

It is not supervised learning:

- There is not data available from an external expert

It is not unsupervised learning:

- It is not about finding structures in data or interpreting unlabeled data.

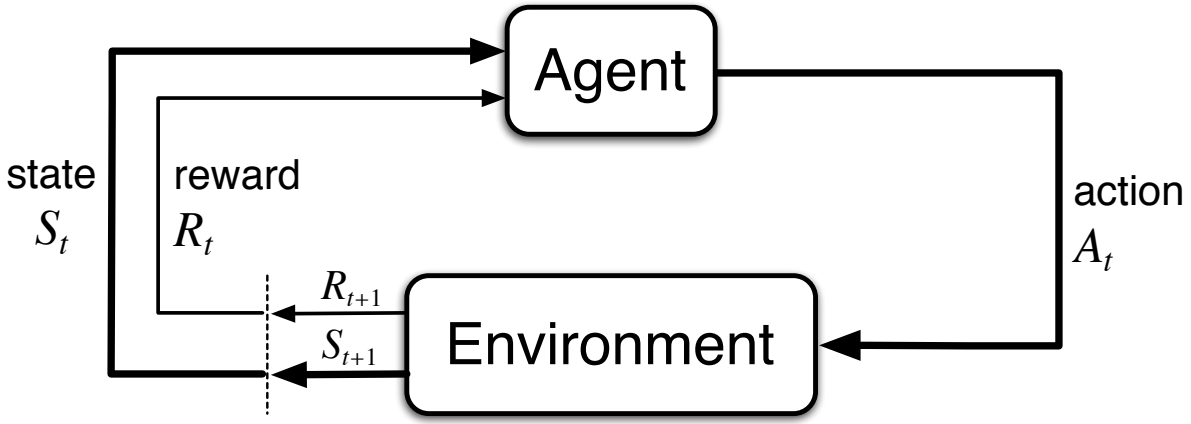
Goal of Reinforcement Learning

In reinforcement learning:

- An agent tries different **actions** and
- Receives a **reward**

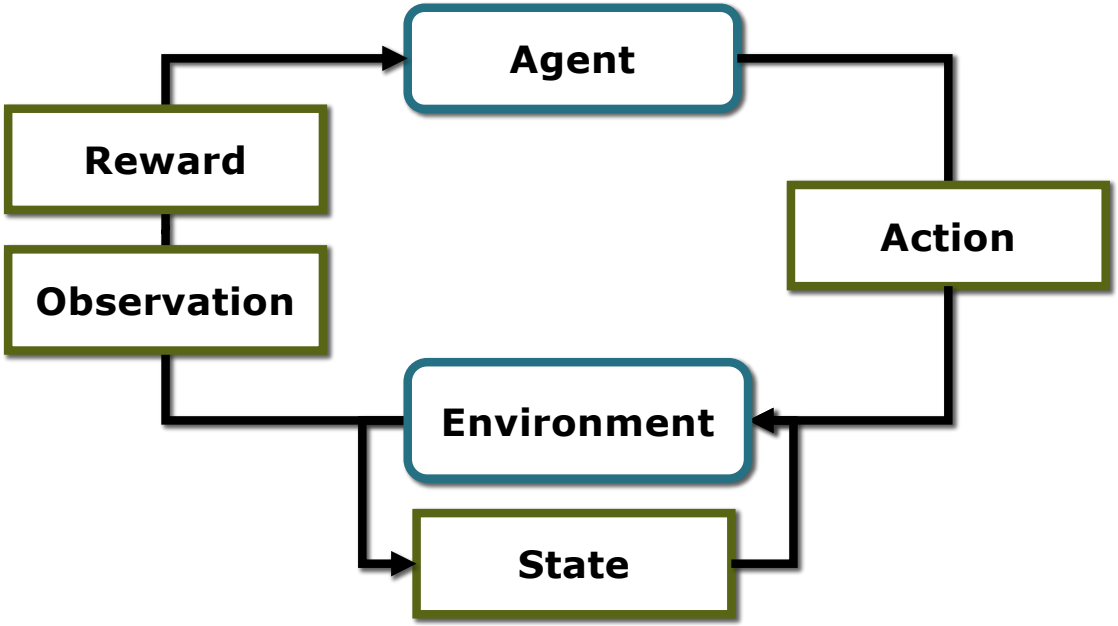
All goals can be described by the maximization of the expected cumulative reward

Agent and Environment



from Sutton & Barto, 2018

Agent and Environment (II)



In many problems and also in the most common implementations, the agent might not receive the full state, but a so called **observation** of the state.

Either for simplicity or because the agent is not able to observe the full state (for example in a game like Poker or Jass)

Cumulative Rewards

Maximizing the cumulative reward or expected return:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

Often, a *discounted* return is used:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$0 \leq \gamma \leq 1$$

Concepts and Notation

A_t	action at time t
S_t	state at time t, typically due, stochastically, to S_{t-1} and A_{t-1}
R_t	reward at time t, typically due, stochastically, to S_{t-1} and A_{t-1}
π	policy (decision-making rule)
$\pi(s)$	action taken in state s under *deterministic* policy π
$\pi(a\|s)$	probability of taking action a in state s under *stochastic* policy π
G_t	return following time t
$v_\pi(s)$	value of state s under policy π (expected return)
$v_*(s)$	value of state s under the optimal policy
$q_\pi(s,a)$	value of taking action a in state s under policy π
$q_*(s,a)$	value of taking action a in state s under the optimal policy

Types of RL Agents (will be covered in the lecture)

Value based:
- No Policy (implicit)
- Value Function

Policy Based:
- Policy
- No Value Function

Actor Critic
- Policy
- Value Function

Model Free
- Policy and/or Value Function
- No Model

Model
- Policy and/or Value Function
- Model (explicit or learned)

Tabular Methods
- Policy and/or Value Function for each state