

# Reinforcement Learning

---

## Exercise 2: Dynamic Programming, Gym & Prediction Problem

Sebastian Rietsch

# Exercise Sheet 2

## Discussion



# MC Prediction



# OpenAI Gym



# Gymnasium

## Intro

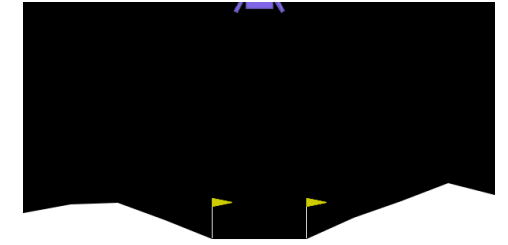
---

- Up to 2021, OpenAI Gym (<https://github.com/openai/gym>) was the industry standard API for defining RL environments
  - Breaking backward compatibility with update v0.26.0
    - “Step” function returns 5 instead of 4 values (termination/truncation)
    - “Reset” function returns 2 values instead of 1 (info)
    - “Seed” function was removed
      - Now: “Reset(seed=...)”
    - “Render” function does not take in render option arguments anymore
  - This might be important if you try running research code from GitHub!
- Has been replaced by Gymnasium (<https://github.com/Farama-Foundation/Gymnasium>) since then
  - See <https://gymnasium.farama.org/> for in-depth documentation
- **We will build all our exercises on Gymnasium v0.28.1 (the most recent version)**



# Gymnasium

## Agent-Environment Interaction Loop



```
import gymnasium as gym
env = gym.make("LunarLander-v2", render_mode="human")
observation, info = env.reset(seed=42)
for _ in range(1000):
    action = env.action_space.sample() # this is where you would insert your policy
    observation, reward, terminated, truncated, info = env.step(action)

    if terminated or truncated:
        observation, info = env.reset()
env.close()
```

# Gymnasium

gymnasium.Env

---

- 4 core methods:
  - “reset(...)”
    - Resets the environment to one of the initial states
    - **Returns:** (initial\_observation, info) tuple
  - “step(action)”:
    - Executes action inside the environment
    - **Returns:** (next\_observation, reward, terminal, truncated, info) tuple
  - “render()”:
    - Render the current state of the environment (if possible)
    - Has to be called at every step
  - “close()”:
    - Close resources used by the environment (free memory, etc.)

# Gymnasium

gymnasium.spaces

---

- Action and observation spaces are of type "Space"
- Spaces
  - "Box": (multi-dimensional) continuous interval(s) from low to high
  - "Discrete": n discrete values
  - "MultiDiscrete"
  - "MultiBinary"
- gymnasium.Env implements 2 core attributes:
  - "action\_space" defines the action space
  - "observation\_space" defines the observation space
    - Both of type "Space"



**Thank you for your attention!**