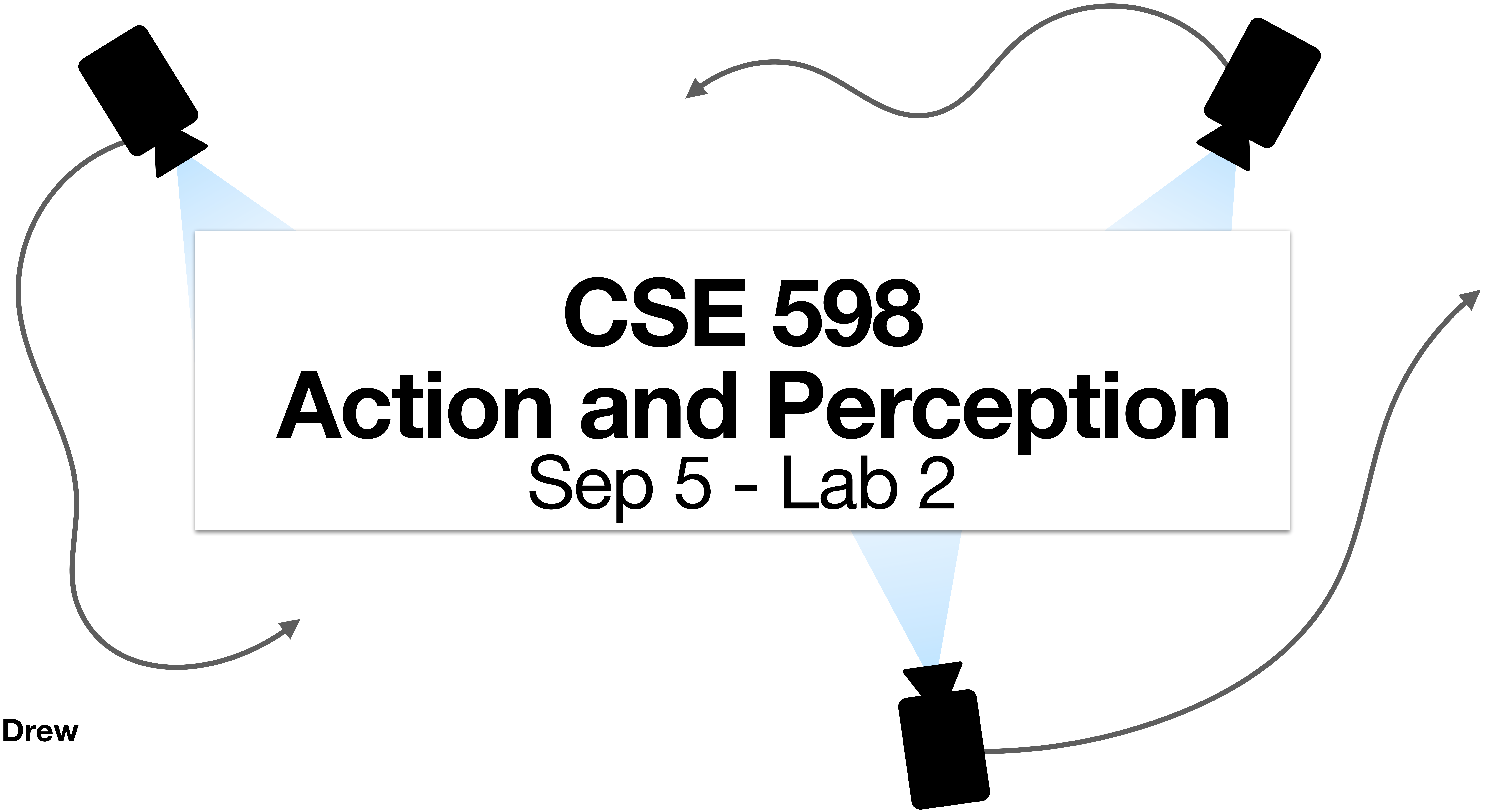
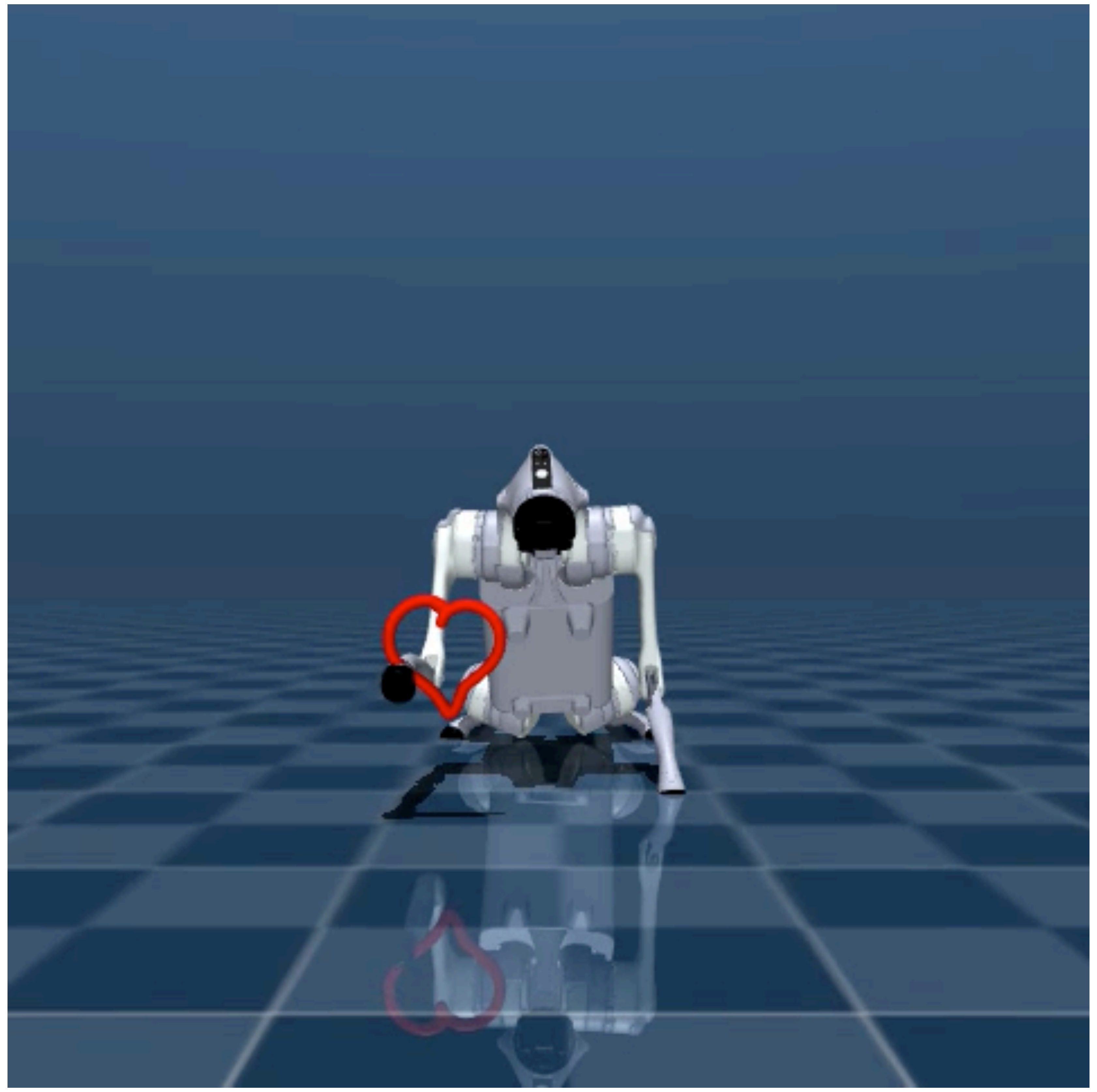


CSE 598
Action and Perception

Sep 5 - Lab 2

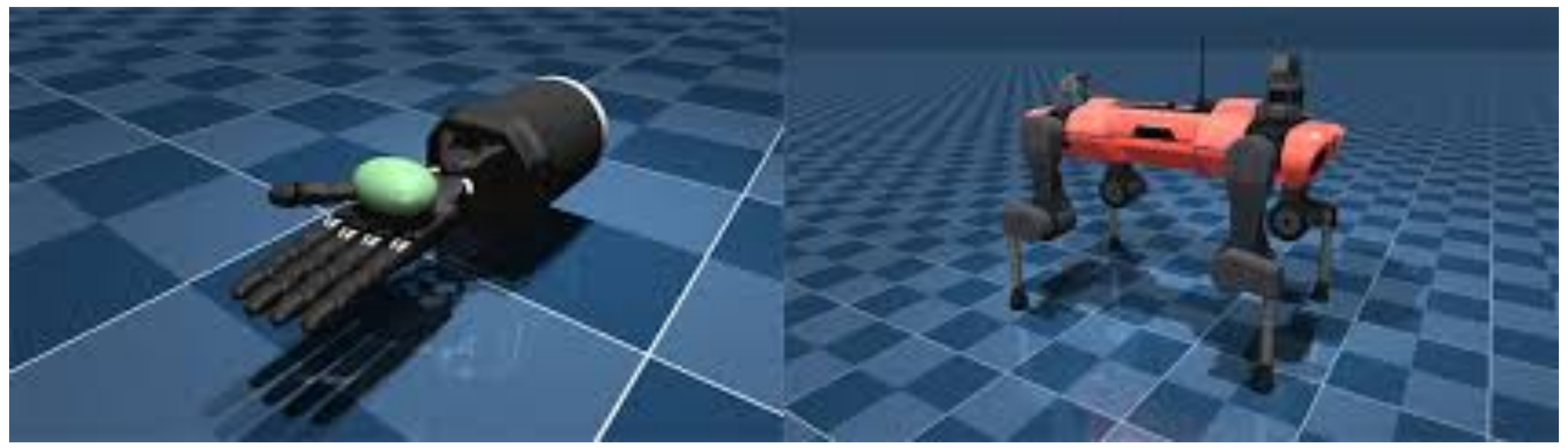
Drew

Basic Environments, Action Spaces, and Rendering


Mujoco Playground Sim

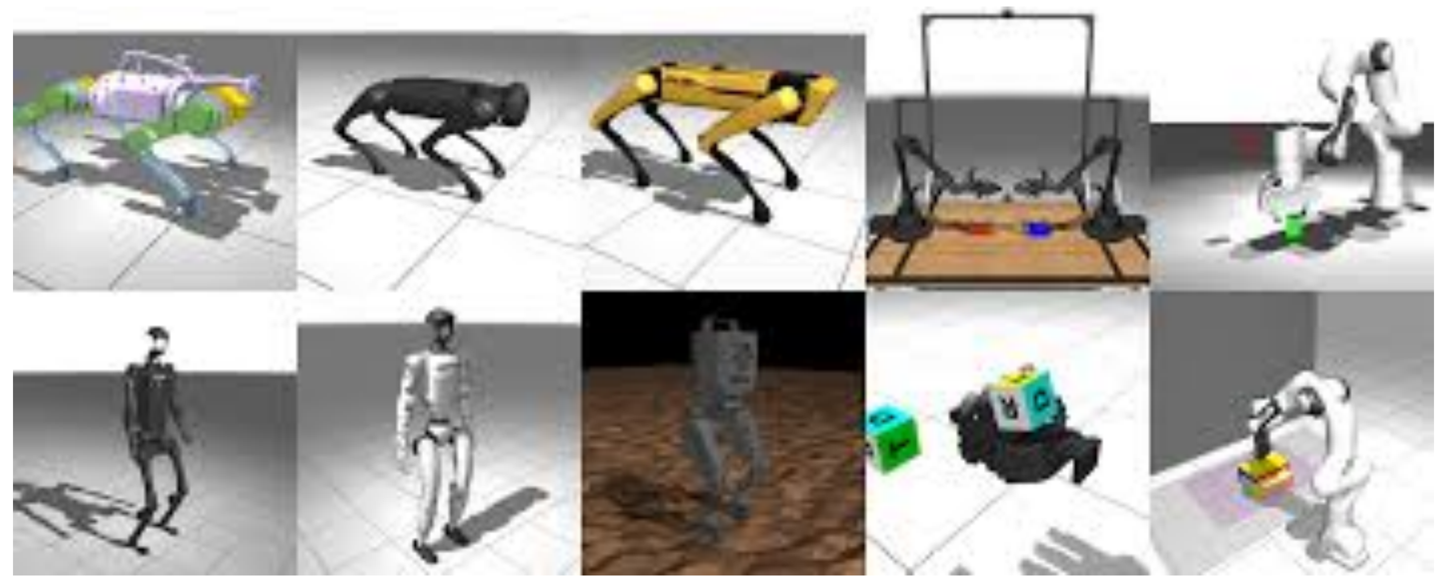
New Simulator: Mujoco Playground

Mujoco



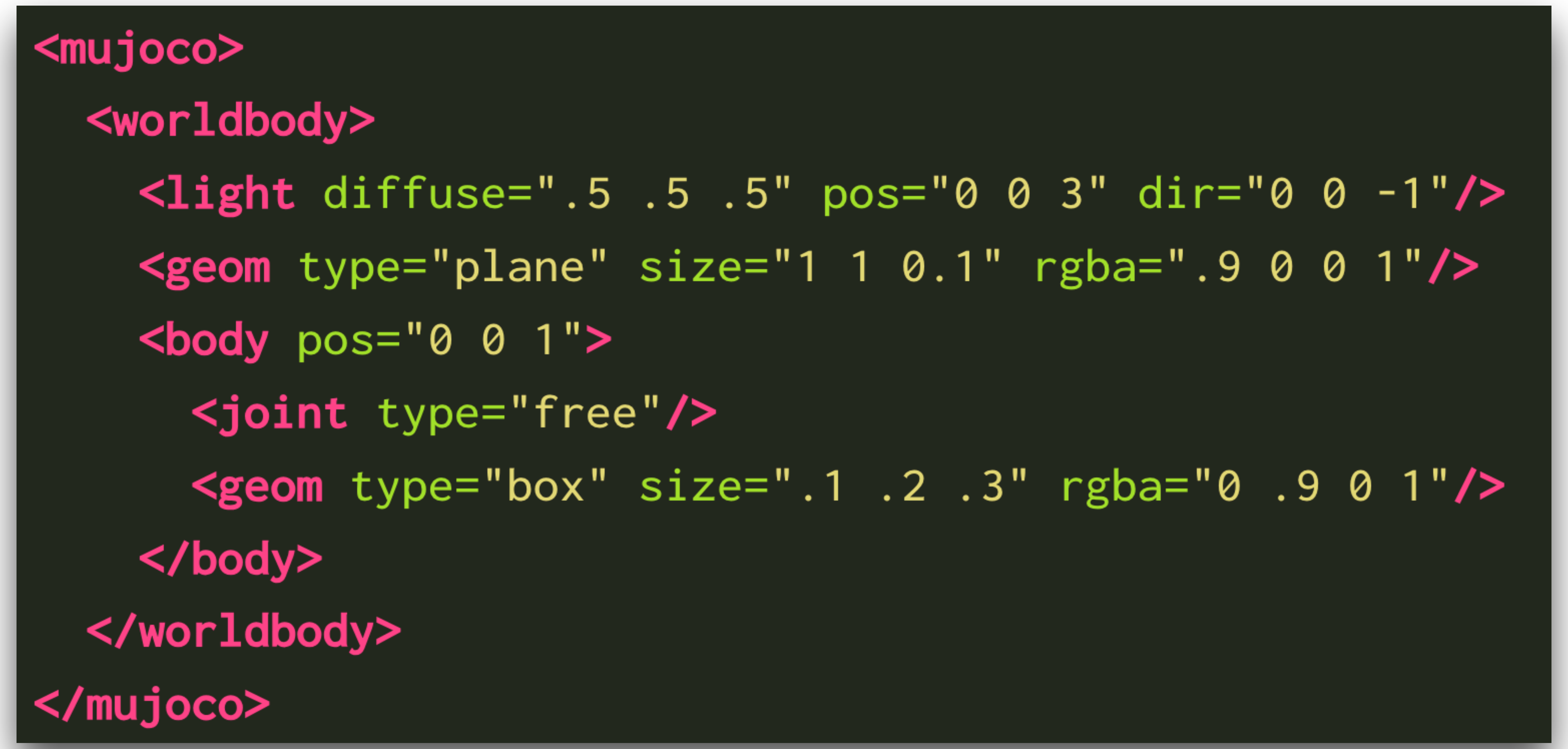
- General physics simulator, designed for high contact and many actuators.
- CPU based

Mujoco Playground



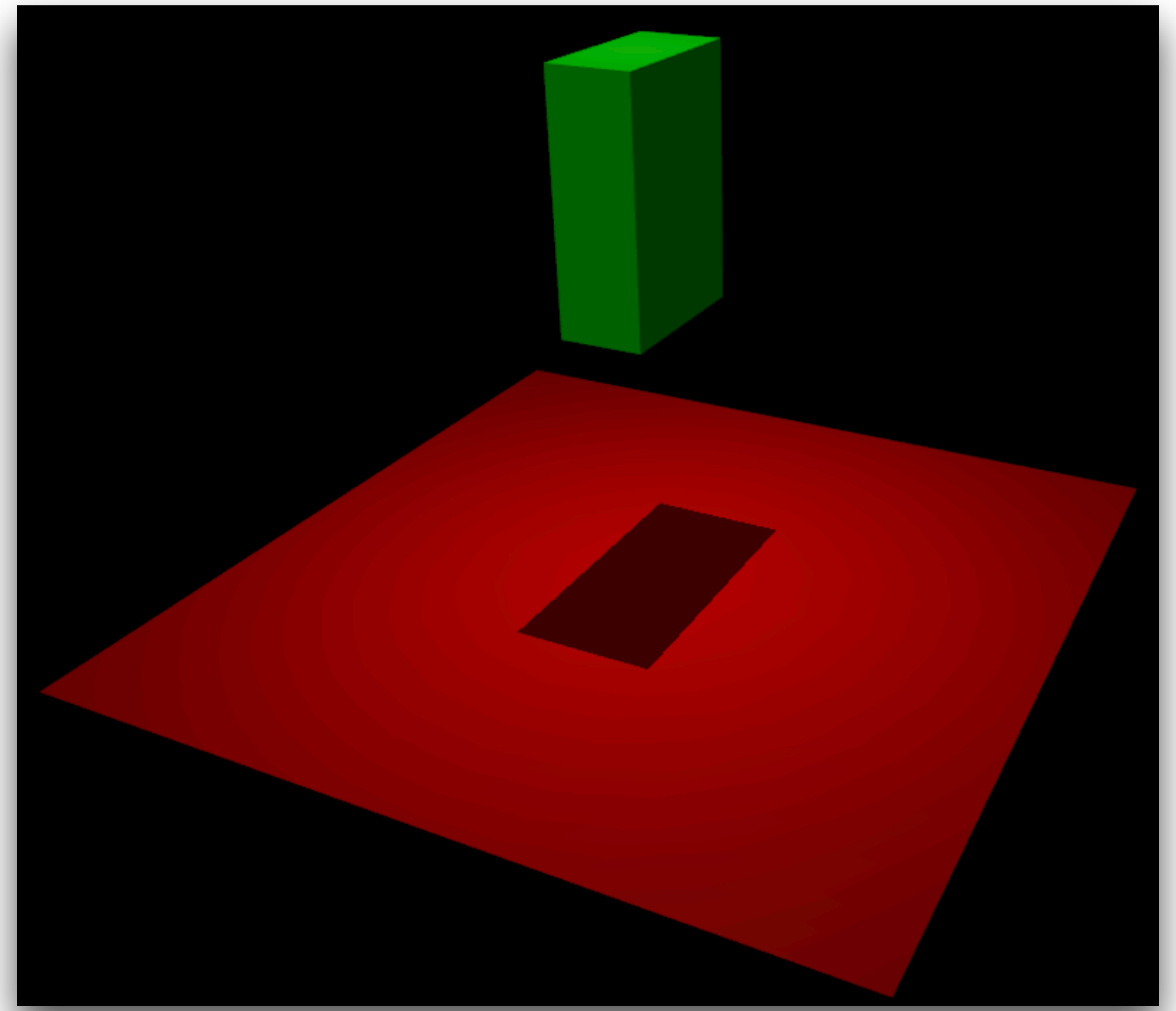
- Reimplimentation of MuJoCo in JAX for highly parallelized RL training
- Standardized environment interfaces for many robots for "quick sim2real transfer"

Description Files



```
<mujoco>
  <worldbody>
    <light diffuse=".5 .5 .5" pos="0 0 3" dir="0 0 -1"/>
    <geom type="plane" size="1 1 0.1" rgba=".9 0 0 1"/>
    <body pos="0 0 1">
      <joint type="free"/>
      <geom type="box" size=".1 .2 .3" rgba="0 .9 0 1"/>
    </body>
  </worldbody>
</mujoco>
```

$\longrightarrow$

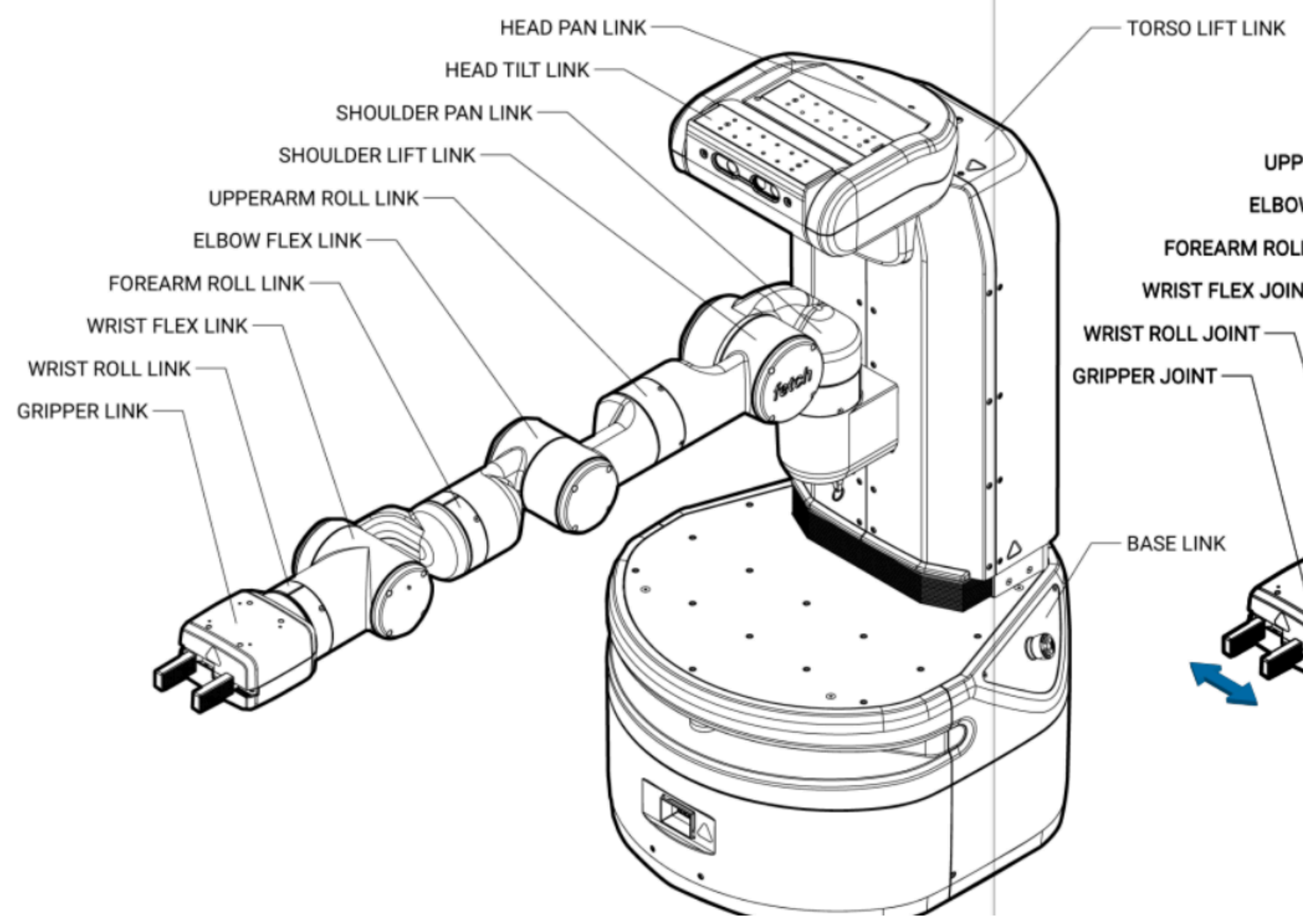


MuJoCo MJCF

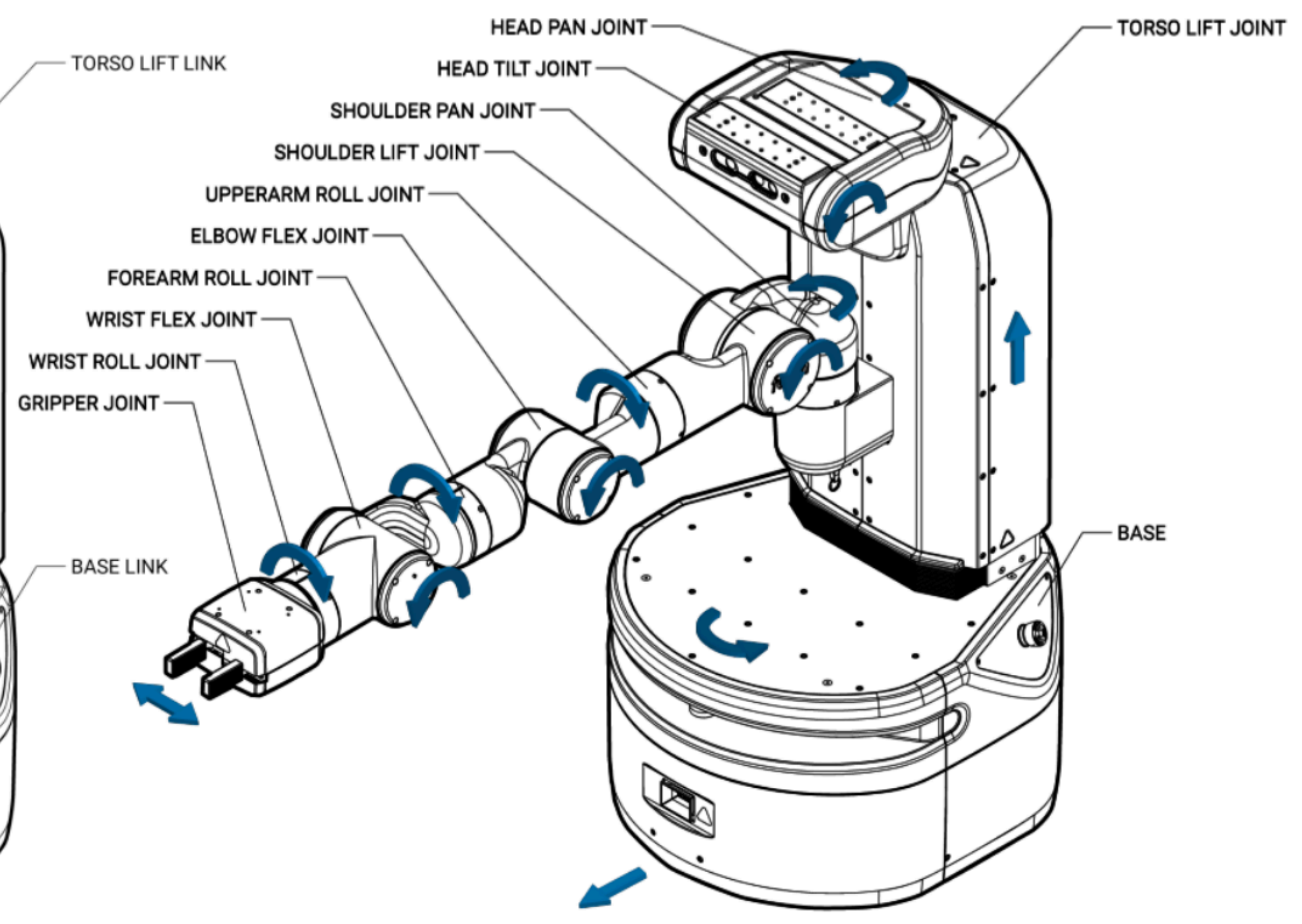
Rendered Scene

Links and Joints

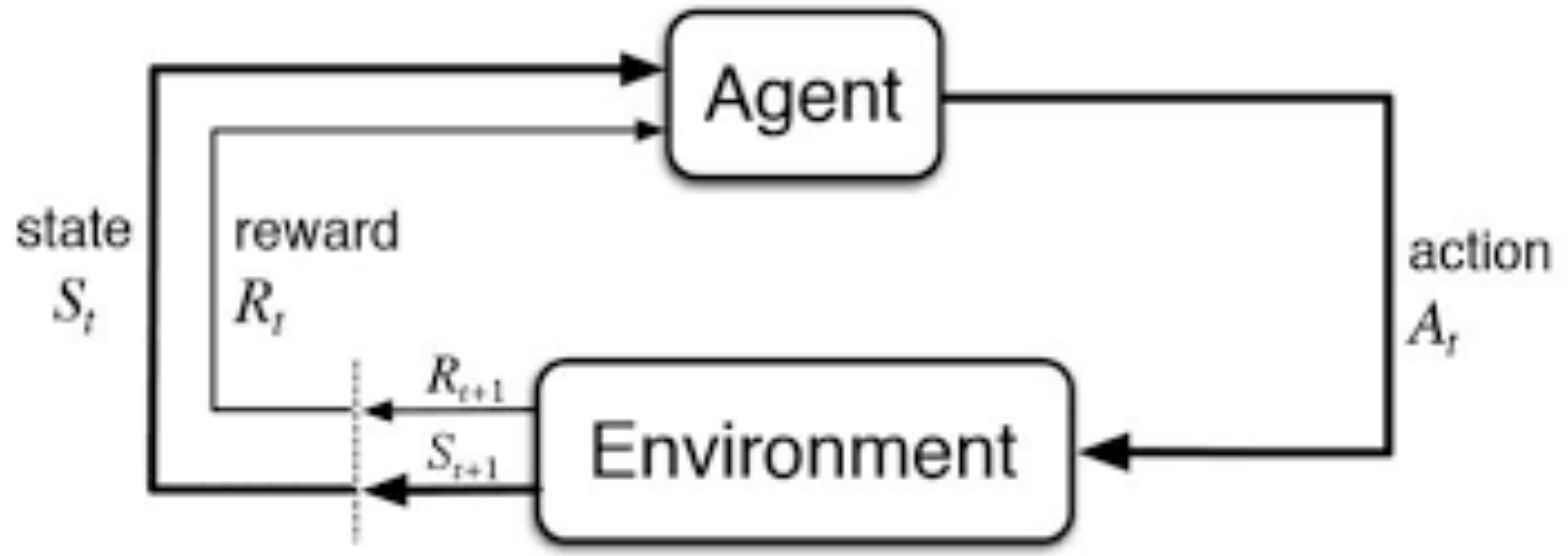
Fetch links



Fetch joints



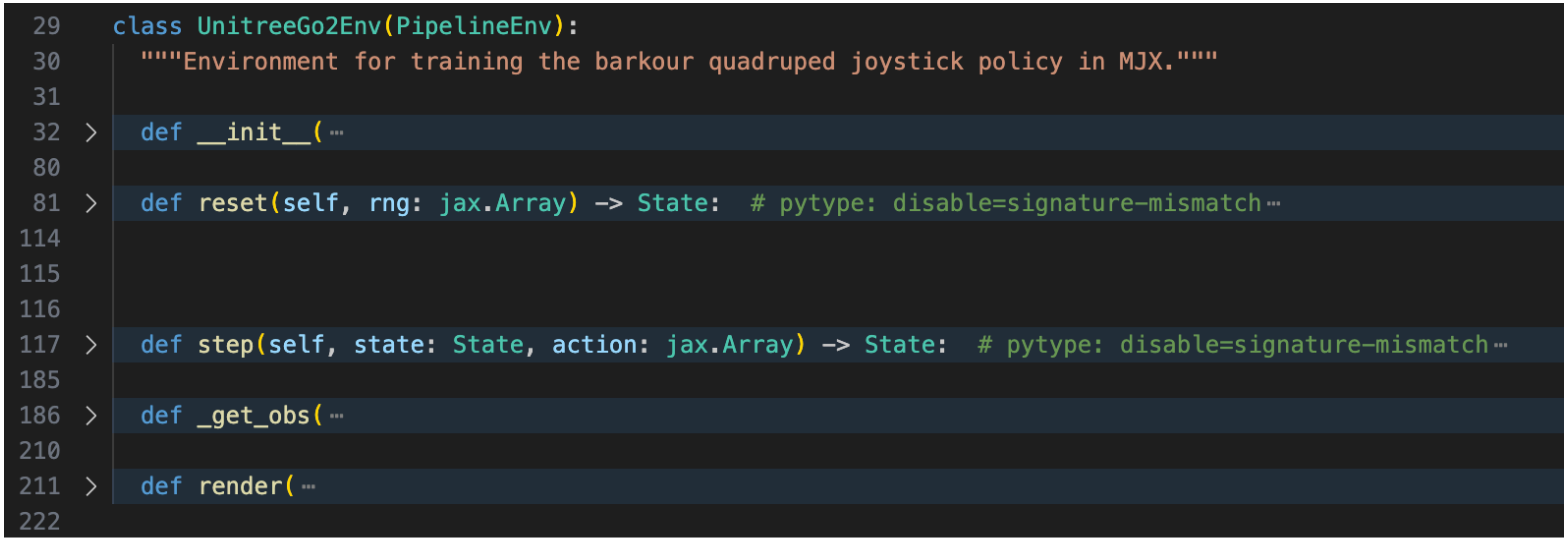
The Standard RL Abstraction



One loop is often called an environment **step()**

Important Environment Attributes

The simulation environment is implemented as a python class



```
class UnitreeGo2Env(PipelineEnv):
  """Environment for training the barkour quadruped joystick policy in MJX."""

  def __init__(…

  def reset(self, rng: jax.Array) -> State:  # pytype: disable=signature-mismatch…



  def step(self, state: State, action: jax.Array) -> State:  # pytype: disable=signature-mismatch…

  def _get_obs(…

  def render(…

```

```
state = State(pipeline_state, obs, reward, done, metrics, state_info)
```

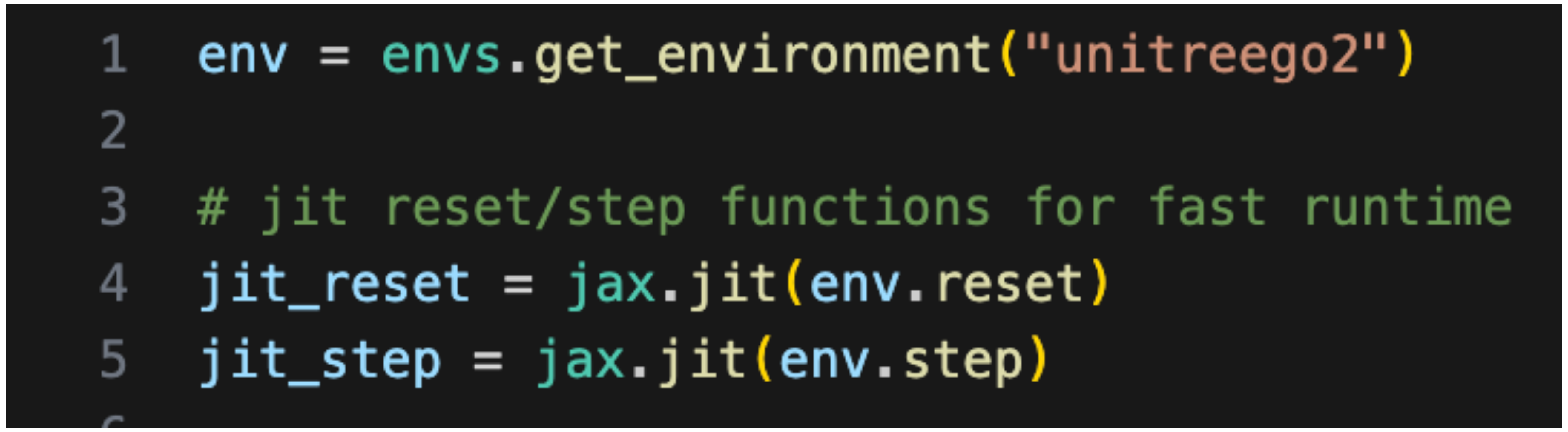
sim-state (pos & vel of every body, etc)

What the "RL agent" would have access to

custom state info (time since kick, last_action, etc)

Just-In-Time (JIT) Compilation w/ MJX

- Enables compilation of MuJoCo’s physics computations into optimized machine code at runtime



```
env = envs.get_environment("unitreego2")

# jit reset/step functions for fast runtime
jit_reset = jax.jit(env.reset)
jit_step = jax.jit(env.step)
```

- Shouldn’t require much effort today. But the compilation changes how random numbers work and also how branching must be done.

Let's Get Started!

- Please bear with us on the grading scheme.
- Plenty of things can go wrong with Google Colab
- **Don't want you to stress about time pressure.** Just get as far as you can by 8:30.

Due by **tomorrow at 1pm**

Quiz Time!

Go to Gradescope and take the **Lab02 Quiz**

Entry Code is **42W5EJ**

Thanks!