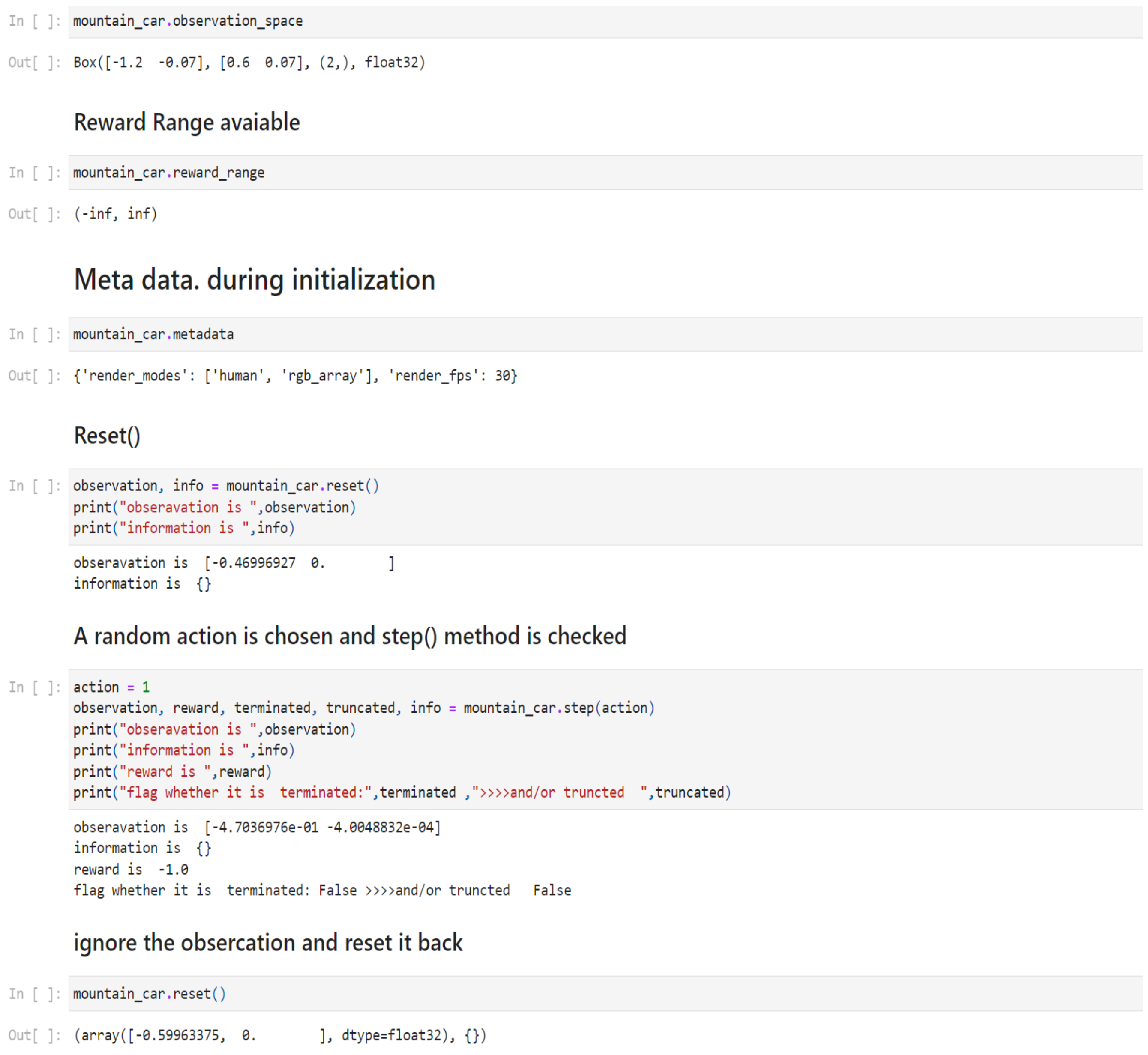
CSE 4/546: Reinforcement Learning
Spring 2023
Quiz 7

Ijaj Ahmed (ijajahme@buffalo.edu)
UB Id: ijajahme

1. The mountain Car environment is imported with gym.make() api to explore other attributes and methods available in gymnasium



```
In [ ]: mountain_car.observation_space

Out[ ]: Box([-1.2  -0.07], [0.6  0.07], (2,), float32)
```

Reward Range avaiable

```
In [ ]: mountain_car.reward_range

Out[ ]: (-inf, inf)
```

Meta data. during initialization

```
In [ ]: mountain_car.metadata

Out[ ]: {'render_modes': ['human', 'rgb_array'], 'render_fps': 30}
```

Reset()

```
In [ ]: observation, info = mountain_car.reset()
        print("obseravation is ",observation)
        print("information is ",info)

        obseravation is  [-0.46996927  0.        ]
        information is  {}
```

A random action is chosen and step() method is checked

```
In [ ]: action = 1
        observation, reward, terminated, truncated, info = mountain_car.step(action)
        print("obseravation is ",observation)
        print("information is ",info)
        print("reward is ",reward)
        print("flag whether it is  terminated:",terminated ,">>>>and/or truncted  ",truncated)

        obseravation is  [-4.7036976e-01 -4.0048832e-04]
        information is  {}
        reward is  -1.0
        flag whether it is  terminated: False >>>>and/or truncted   False
```

ignore the obsercation and reset it back

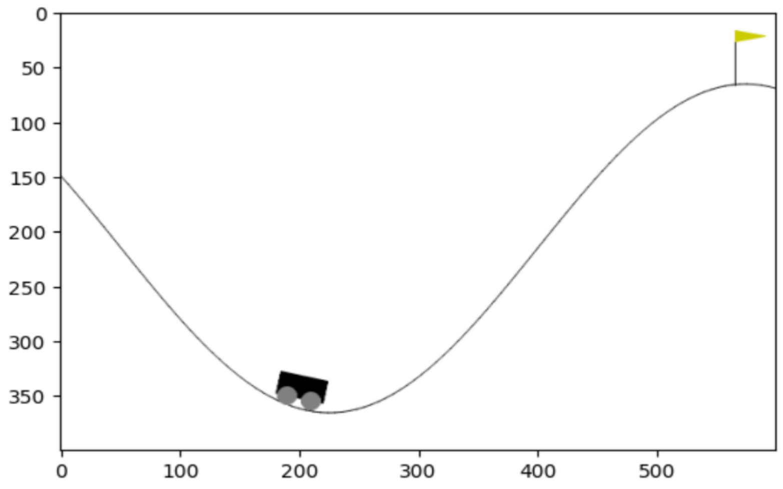
```
In [ ]: mountain_car.reset()

Out[ ]: (array([-0.59963375,  0.        ], dtype=float32), {})
```

Again re-initialized with metadata to check the render() method

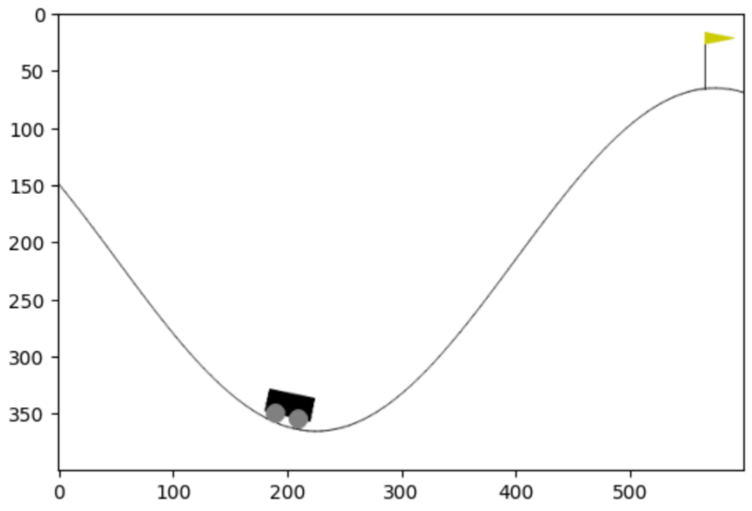
In [ ]:
```
mountain_car=gym.make('MountainCar-v0',render_mode="rgb_array")
mountain_car.reset()

envrend=mountain_car.render()
plt.imshow(envrend)
plt.show()
```



In [ ]:
```
action = 1
observation, reward, terminated, truncated, info = mountain_car.step(action)
envrend=mountain_car.render()
plt.imshow(envrend)
plt.show()
```

In [ ]:
```
action = 1
observation, reward, terminated, truncated, info = mountain_car.step(action)
envrend=mountain_car.render()
plt.imshow(envrend)
plt.show()
```



Rendering close method

In [ ]:
```
mountain_car.close()
```

2. Explore ‘CartPole-v1’

Describing the CartPole environment

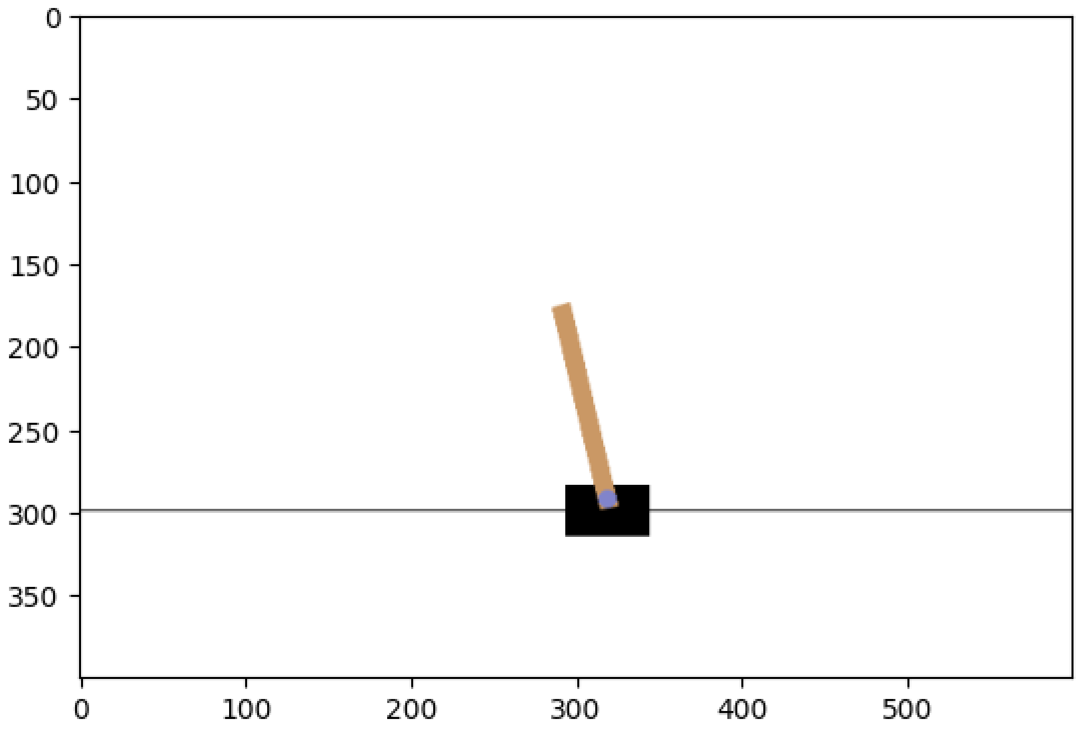
A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.

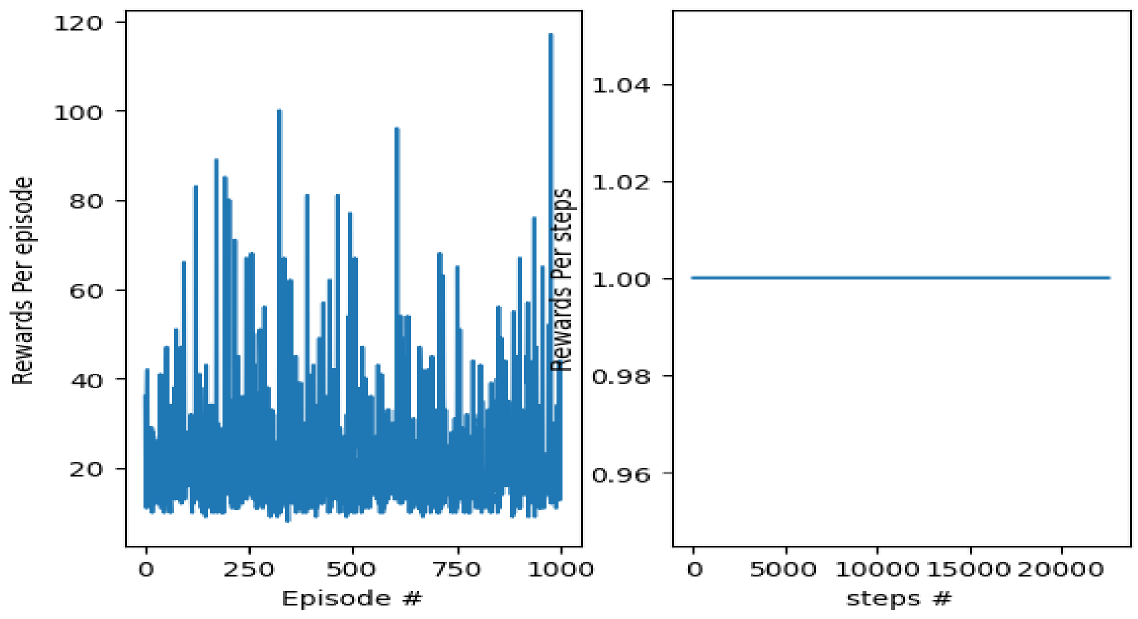
Action: It has two action 0/1 to push to left and right respectively

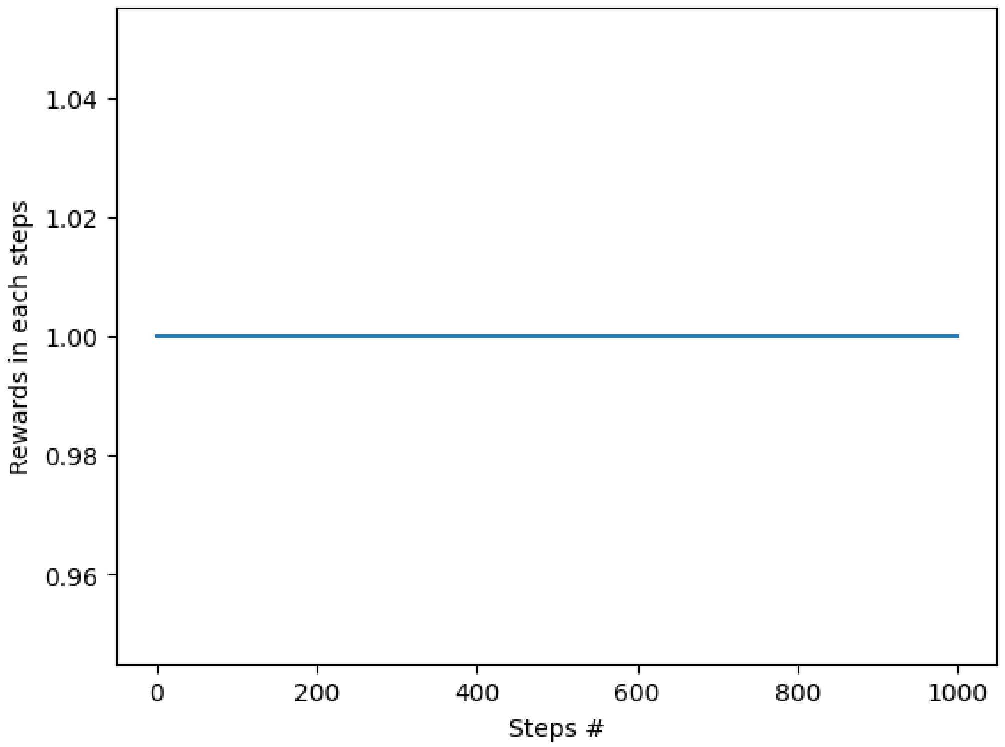
Observation: It has 4 observation (a) Cart position (b) Cart velocity (c) Pole Angle and (d) Pole Angular velocity

Reward: a reward of +1 for every step taken, including the termination step, is allotted. For every step 1 reward is added till the termination or truncation occurred. The idea is to remain in a stable position as long as possible.

As a result in each episode, the cumulative reward is equal to the duration of the episode.







A reward of +1 for every step taken, including the termination step, is allotted. So the reward is observed to be 1 for all the steps. However in each episode the reward reflects the duration the pole is in stable position respecting the boundary or range spec set by the gymnasium library for cartpole environment

3 Exploring acrobot environment

Describing the acrobot environment

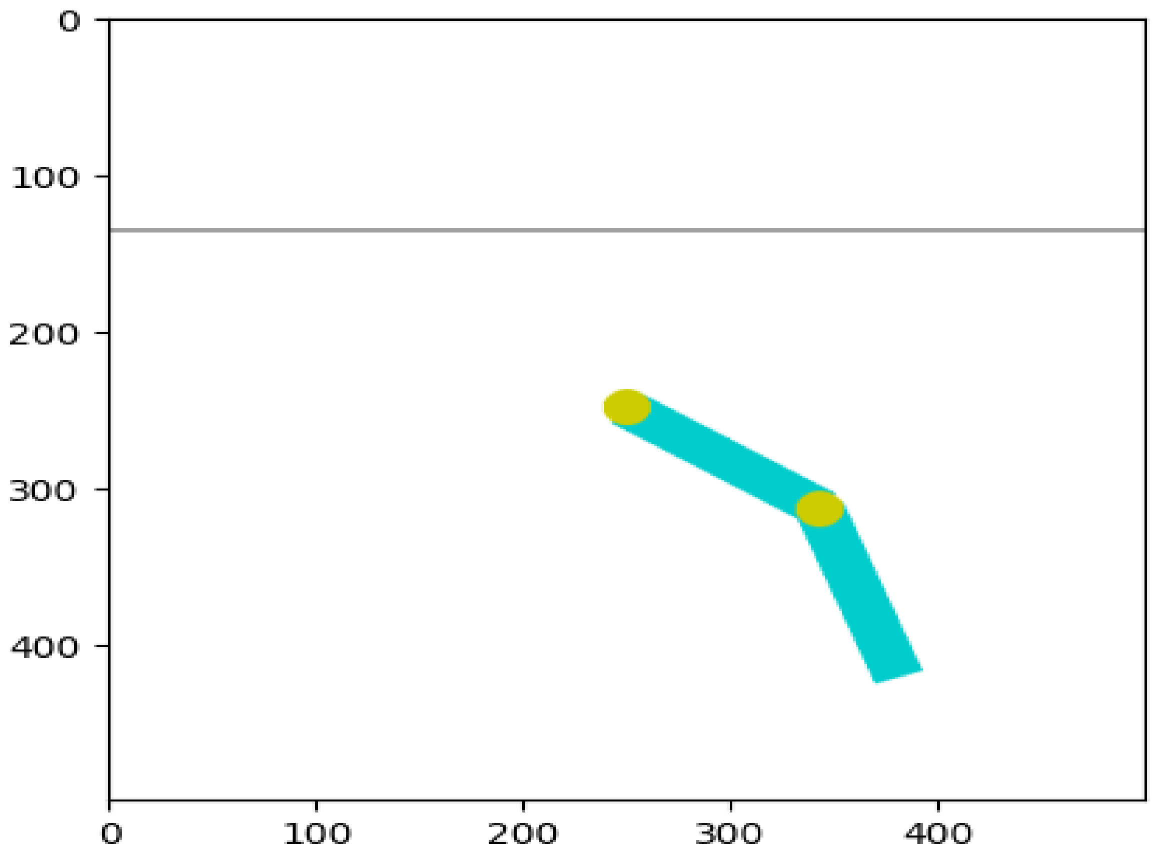
The system consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards.

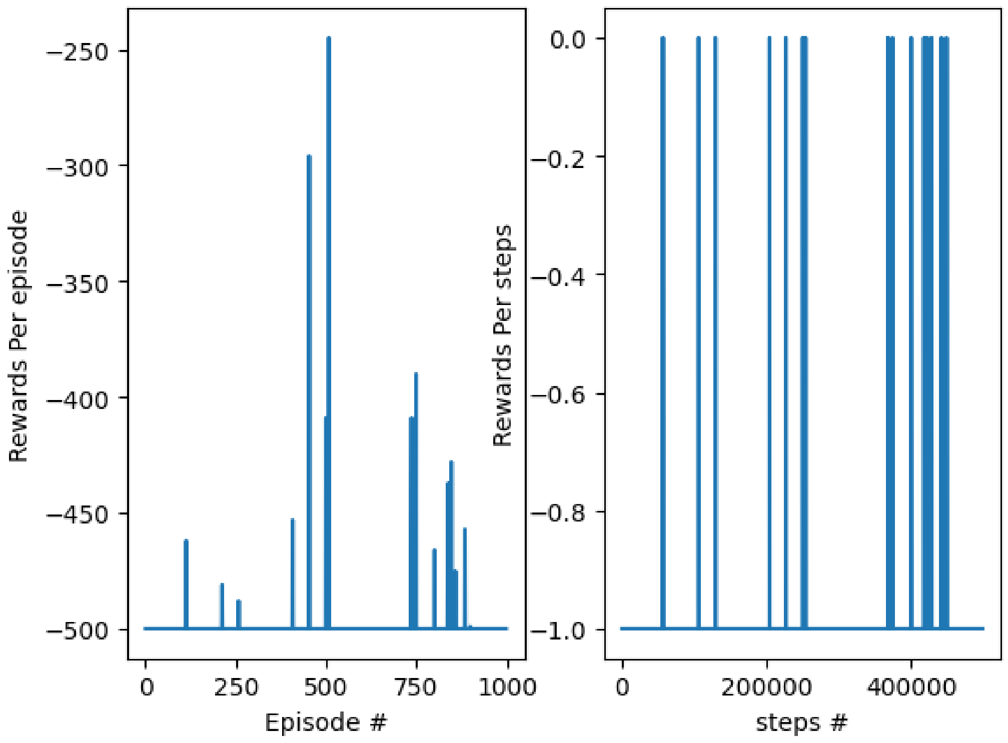
Action: The action space is three and represents the torque applied on the actuated joint between the two links.0,1 and 2 represents -1 N, 0 N and 1 N respectively torque applied at the joint

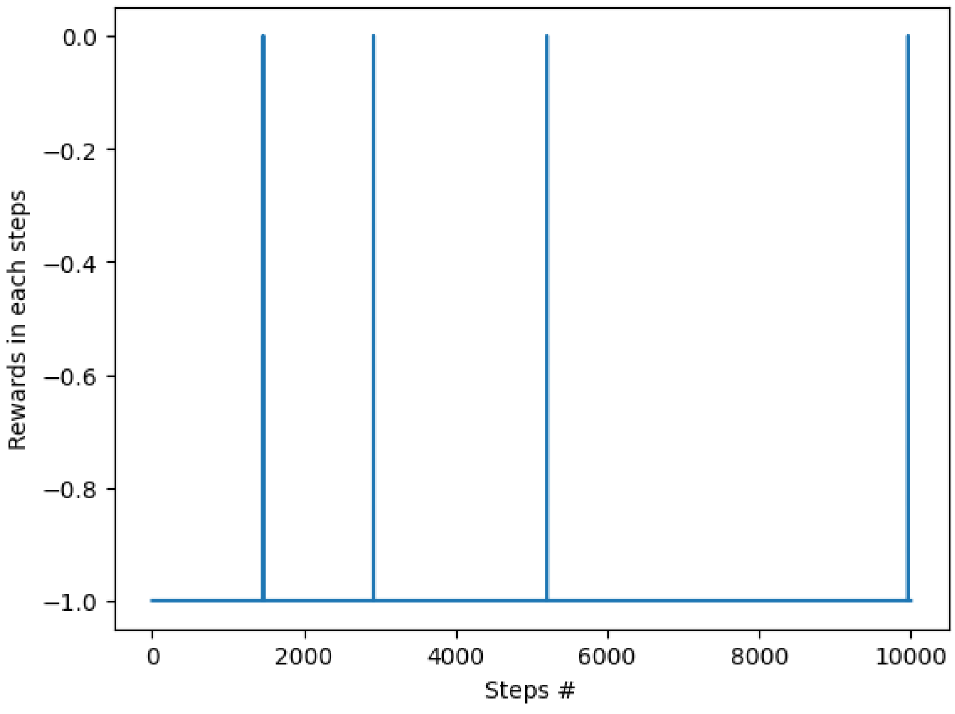
Observation : The observation is a ndarray with shape (6,) that provides information about the two rotational joint angles as well as their angular velocities

Rewards : the goal is to have the free end reach a designated target height in as few steps as possible, and as such all steps that do not reach the goal incur a reward of -1. Achieving the target height results in termination with a reward of 0.

The negative ensures that the target is achieved in a minimum number of steps . At each step until it reaches the terminal state -1 is penalized and for each episode the cumulative reward is negative of the number of steps plus 1. The last step is awarded no penally







The reward is negative in each step . Only when it reaches the final step the reward is zero. Here the steps are shown for all the episodes with each episode running till terminated or truncated by the out of the range condition. The spike of 0 reward in some cases show the terminal state is achieved in few cases without violating the other termination condition