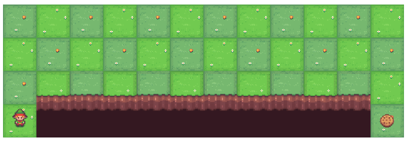
Cliff Walking



This environment is part of the Toy Text environments which contains general information about the environment.

Action Space	`Discrete(4)`
Observation Space	`Discrete(48)`
import	`gymnasium.make("CliffWalking-v0")`

Cliff walking involves crossing a gridworld from start to goal while avoiding falling off a cliff.

Description

The game starts with the player at location [3, 0] of the 4x12 grid world with the goal located at [3, 11]. If the player reaches the goal the episode ends.

A cliff runs along [3, 1..10]. If the player moves to a cliff location it returns to the start location.

The player makes moves until they reach the goal.

Adapted from Example 6.6 (page 132) from Reinforcement Learning: An Introduction by Sutton and Barto [1].

The cliff can be chosen to be slippery (disabled by default) so the player may move perpendicular to the intended direction sometimes (see `is_slippery`).

With inspiration from: https://github.com/dennybritz/reinforcement-learning/blob/master/lib/envs/cliff_walking.py

Action Space

The action shape is `(1,)` in the range `{0, 3}` indicating which direction to move the player.

- 0: Move up
- 1: Move right
- 2: Move down
- 3: Move left

v1.1.1 (latest)

There are 3 x 12 + 1 possible states. The player cannot be at the cliff, nor at the goal as the latter results in the end of the episode. What remains are all the positions of the first 3 rows plus the bottom-left cell.

The observation is a value representing the player’s current position as current_row * ncols + current_col (where both the row and col start at 0).

For example, the starting position can be calculated as follows: 3 * 12 + 0 = 36.

The observation is returned as an `int()`.

Starting State

The episode starts with the player in state `[36]` (location [3, 0]).

Reward

Each time step incurs -1 reward, unless the player stepped into the cliff, which incurs -100 reward.

Episode End

The episode terminates when the player enters state `[47]` (location [3, 11]).

Information

`step()` and `reset()` return a dict with the following keys:

- “p” - transition proability for the state.

As cliff walking is not stochastic, the transition probability returned always 1.0.

Arguments

```
import gymnasium as gym
gym.make('CliffWalking-v1')
```

References

[1] R. Sutton and A. Barto, “Reinforcement Learning: An Introduction” 2020. [Online]. Available: http://www.incompleteideas.net/book/RLbook2020.pdf

- v1: Add slippery version of cliffwalking
- v0: Initial version release