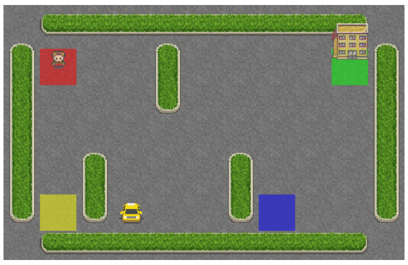
Taxi



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

Action Space	`Discrete(6)`
Observation Space	`Discrete(500)`
import	`gymnasium.make("Taxi-v3")`

The Taxi Problem involves navigating to passengers in a grid world, picking them up and dropping them off at one of four locations.

Description

There are four designated pick-up and drop-off locations (Red, Green, Yellow and Blue) in the 5x5 grid world. The taxi starts off at a random square and the passenger at one of the designated locations.

The goal is move the taxi to the passenger's location, pick up the passenger, move to the passenger's desired destination, and drop off the passenger. Once the passenger is dropped off, the episode ends.

The player receives positive rewards for successfully dropping-off the passenger at the correct location. Negative rewards for incorrect attempts to pick-up/drop-off passenger and for each step where another reward is not received.

Map:

```
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

From “Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition” by Tom Dietterich [1].

Action Space

The action shape is `(1,)` in the range `{0, 5}` indicating which direction to move the taxi or to pickup/drop off passengers.

- 0: Move south (down)
- 1: Move north (up)
- 2: Move east (right)
- 3: Move west (left)
- 4: Pickup passenger
- 5: Drop off passenger

Observation Space

There are 500 discrete states since there are 25 taxi positions, 5 possible locations of the passenger (including the case when the passenger is in the taxi), and 4 destination locations.

Destination on the map are represented with the first letter of the color.

Passenger locations:

- 0: Red
- 1: Green
- 2: Yellow
- 3: Blue
- 4: In taxi

Destinations:

- 0: Red
- 1: Green
- 2: Yellow
- 3: Blue

Note that there are 400 states that can actually be reached during an episode. The missing states correspond to situations in which the passenger is at the same location as their destination, as this typically signals the end of an episode. Four additional states can be observed right after a successful episodes, when both the passenger and the taxi are at the destination. This gives a total of 404 reachable discrete states.

Starting State

The initial state is sampled uniformly from the possible states where the passenger is neither at their destination nor inside the taxi. There are 300 possible initial states: 25 taxi positions, 4 passenger locations (excluding inside the taxi) and 3 destinations (excluding the passenger's current location).

Rewards

- -1 per step unless other reward is triggered.
- +20 delivering passenger.
- -10 executing "pickup" and "drop-off" actions illegally.

An action that results a noop, like moving into a wall, will incur the time step penalty. Noops can be avoided by sampling the `action_mask` returned in `info`.

Episode End

The episode ends if the following happens:

- Termination: 1. The taxi drops off the passenger.
- Truncation (when using the time_limit wrapper): 1. The length of the episode is 200.

Information

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

- p - transition proability for the state.
- action_mask - if actions will cause a transition to a new state.

As taxi is not stochastic, the transition probability is always 1.0. Implementing a transitional probability in line with the Dietterich paper ('The fickle taxi task') is a TODO.

For some cases, taking an action will have no effect on the state of the episode. In v0.25.0, `info["action_mask"]` contains a np.ndarray for each of the actions specifying if the action will

To sample a modifying action, use `action = env.action_space.sample(info["action_mask"])` Or with a Q-value based algorithm `action = np.argmax(q_values[obs, np.where(info["action_mask"] == 1)[0]])`.

Arguments

```python
import gymnasium as gym
gym.make('Taxi-v3')
```

References

[1] T. G. Dietterich, "Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition," Journal of Artificial Intelligence Research, vol. 13, pp. 227–303, Nov. 2000, doi: 10.1613/jair.639.

Version History

- v3: Map Correction + Cleaner Domain Description, v0.25.0 action masking added to the reset and step information
- v2: Disallow Taxi start location = goal location, Update Taxi observations in the rollout, Update Taxi reward threshold.
- v1: Remove (3,2) from locs, add passidx<4 check
- v0: Initial version release