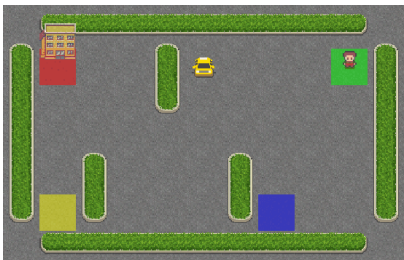


# de taxi



Este entorno forma parte de los entornos Toy Text que contiene información general sobre el entorno.

Espacio de acción	Discrete(6)
Espacio de observación	Discrete(500)
importar	<code>gymnasium.make("Taxi-v3")</code>

The Taxi Problem implica navegar hasta los pasajeros en un mundo en red, recogerlos y dejarlos en una de cuatro ubicaciones.

## Descripción

Hay cuatro lugares designados para recoger y dejar (rojo, verde, amarillo y azul) en el mundo de la cuadrícula 5x5. El taxi sale de una plaza aleatoria y el pasajero de uno de los lugares designados.

El objetivo es mover el taxi a la ubicación del pasajero, recoger al pasajero, trasladarse al destino deseado del pasajero y dejar al pasajero. Una vez que se deja al pasajero, el episodio termina.

El jugador recibe recompensas positivas por dejar con éxito al pasajero en el lugar correcto. Recompensas negativas por intentos incorrectos de recoger/dejar a un pasajero y por cada paso en el que no se reciba otra recompensa.

Mapa:

```
+-----+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+-----+
```

v0.29.0 (más reciente)



## Espacio de acción

La forma de acción está `(1,)` en el rango que indica en qué dirección mover el taxi o recoger/dejar pasajeros. `{0, 5}`

- 0: moverse hacia el sur (abajo)
- 1: Muévete hacia el norte (arriba)
- 2: Muévete hacia el este (derecha)
- 3: Muévete hacia el oeste (izquierda)
- 4: pasajero de recogida
- 5: dejar pasajero

## Espacio de observación

Hay 500 estados discretos ya que hay 25 posiciones de taxi, 5 ubicaciones posibles del pasajero (incluido el caso cuando el pasajero está en el taxi) y 4 ubicaciones de destino.

El destino en el mapa se representa con la primera letra del color.

Ubicaciones de pasajeros:

- 0: rojo
- 1: verde
- 2: amarillo
- 3: azul
- 4: En taxi

Destinos:

- 0: rojo
- 1: verde
- 2: amarillo
- 3: azul

Una observación se devuelve como un `int()` que codifica el estado correspondiente, calculado por `((taxi_row * 5 + taxi_col) * 5 + passenger_location) * 4 + destination`

Tenga en cuenta que hay 400 estados a los que se puede llegar durante un episodio. Los estados faltantes corresponden a situaciones en las que el pasajero se encuentra en el mismo lugar que su destino, ya que esto suele señalar el final de un episodio. Se pueden observar cuatro estados

v0.29.0 (más reciente)



## Estado inicial

El episodio comienza con el jugador en un estado aleatorio.

## Recompensas

- -1 por paso a menos que se active otra recompensa.
- +20 entregando pasajero.
- -10 ejecutar acciones de "recogida" y "devolución" de forma ilegal.

Una acción que resulte en un fracaso, como moverse contra una pared, incurrirá en la penalización por paso de tiempo. Se pueden evitar errores probando lo `action_mask` devuelto en formato `info`.

## Fin del episodio

El episodio termina si sucede lo siguiente:

- Terminación: 1. El taxi deja al pasajero.
- Truncamiento (cuando se utiliza el contenedor `time_limit`): 1. La duración del episodio es 200.

## Información

`step()` y `reset()` devolver un dict con las siguientes claves:

- `p` - probabilidad de transición para el estado.
- `action_mask`: si las acciones provocarán una transición a un nuevo estado.

Como el taxi no es estocástico, la probabilidad de transición es siempre 1,0. Implementar una probabilidad de transición en línea con el artículo de Dietterich ("La voluble tarea del taxi") es una tarea TODO.

En algunos casos, realizar una acción no tendrá ningún efecto sobre el estado del episodio. En v0.25.0, `info["action_mask"]` contiene un `np.ndarray` para cada una de las acciones que especifica si la acción cambiará el estado.

Para probar una acción de modificación, utilice `O` con un algoritmo basado en valor `Q`. `action = env.action_space.sample(info["action_mask"]) action = np.argmax(q_values[obs, np.where(info["action_mask"] == 1)[0]])`



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

## Referencias

[1] TG Dietterich, "Aprendizaje por refuerzo jerárquico con la descomposición de la función de valor MAXQ", Journal of Artificial Intelligence Research, vol. 13, págs. 227–303, noviembre de 2000, doi: 10.1613/jair.639.

## Historial de versiones

- v3: Corrección de mapa + Descripción de dominio más limpia, máscara de acción v0.25.0 agregada al restablecimiento y a la información de pasos
- v2: No permitir la ubicación de inicio del taxi = ubicación objetivo, actualizar las observaciones del taxi en el lanzamiento, actualizar el umbral de recompensa del taxi.
- v1: Eliminar (3,2) de las ubicaciones, agregar passidx<4 verificar
- v0: Lanzamiento de la versión inicial



Copyright © 2023 Fundación Farama

