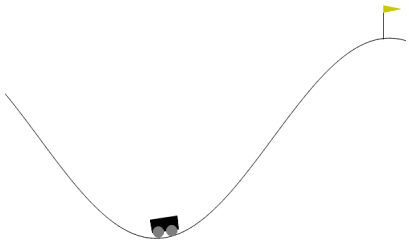
# Mountain Car



This environment is part of the Classic Control environments which contains general information about the environment.

| | |
|---|---|
| Action Space | `Discrete(3)` |
| Observation Space | `Box([-1.2 -0.07], [0.6 0.07], (2,), float32)` |
| import | `gymnasium.make("MountainCar-v0")` |

## Description

The Mountain Car MDP is a deterministic MDP that consists of a car placed stochastically at the bottom of a sinusoidal valley, with the only possible actions being the accelerations that can be applied to the car in either direction. The goal of the MDP is to strategically accelerate the car to reach the goal state on top of the right hill. There are two versions of the mountain car domain in gymnasium: one with discrete actions and one with continuous. This version is the one with discrete actions.

This MDP first appeared in Andrew Moore's PhD Thesis (1990)

```
@TECHREPORT{Moore90efficientmemory-based,
    author = {Andrew William Moore},
    title = {Efficient Memory-based Learning for Robot Control},
    institution = {University of Cambridge},
    year = {1990}
}
```

## Observation Space

The observation is a `ndarray` with shape `(2,)` where the elements correspond to the following:

| | | | | |
|---|---|---|---|---|
| 0 | position of the car along the x-axis | -1.2 | 0.6 | position (m) |
| 1 | velocity of the car | -0.07 | 0.07 | velocity (v) |

## Action Space

There are 3 discrete deterministic actions:

- 0: Accelerate to the left
- 1: Don't accelerate
- 2: Accelerate to the right

## Transition Dynamics:

Given an action, the mountain car follows the following transition dynamics:

$velocity_{t+1} = velocity_t + (action - 1) * force - cos(3 * position_t) * gravity$

$position_{t+1} = position_t + velocity_{t+1}$

where force = 0.001 and gravity = 0.0025. The collisions at either end are inelastic with the velocity set to 0 upon collision with the wall. The position is clipped to the range `[-1.2, 0.6]` and velocity is clipped to the range `[-0.07, 0.07]`.

## Reward:

The goal is to reach the flag placed on top of the right hill as quickly as possible, as such the agent is penalised with a reward of -1 for each timestep.

## Starting State

The position of the car is assigned a uniform random value in *[-0.6 , -0.4]*. The starting velocity of the car is always assigned to 0.

## Episode End

The episode ends if either of the following happens:

1. Termination: The position of the car is greater than or equal to 0.5 (the goal position on top of the right hill)
2. Truncation: The length of the episode is 200.

Mountain Car has two parameters for `gymnasium.make` with `render_mode` and `goal_velocity`. On reset, the `options` parameter allows the user to change the bounds used to determine the new random state.

```
>>> import gymnasium as gym
>>> env = gym.make("MountainCar-v0", render_mode="rgb_array", goal_velocity=0.1)  # default go
>>> env
<TimeLimit<OrderEnforcing<PassiveEnvChecker<MountainCarEnv<MountainCar-v0>>>>>
>>> env.reset(seed=123, options={"x_init": np.pi/2, "y_init": 0.5})  # default x_init=np.pi, y
(array([-0.46352962,  0.        ], dtype=float32), {})
```

# Version History

- v0: Initial versions release