Optional Challenge
Reward Shaping in MOUNTAIN CAR

Learning Objectives

The objective of this challenge is to experiment with Reward Shaping in the Mountain Car environment. You are provided with a complete program that you must modify to make it converge as fast as possible. The learning algorithm is Q-learning.

Information about the environment can be found in [Far25].

1 Files for this Practice

```
021_Q_learning_MOUNTAIN_CAR.ipynb
```

2 Reward Shaping

Reward shaping in reinforcement learning (RL) refers to the technique of modifying or augmenting the reward signal to make learning more efficient or to encourage certain behaviors. It is often used to guide the agent more effectively toward desired policies, especially when the original reward is sparse, delayed, or difficult to optimize.

We should use Reward Shaping for:

1. Speed up convergence: Helps the agent learn useful behaviors faster.
2. Provide intermediate feedback: Useful in sparse-reward environments.
3. Incorporate prior knowledge: Allows human designers to encode hints or goals

3 The Mountain Car environment

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.

The action space and Observation space is as follows:

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. There is no final reward and the episode finishes at the 200 step.

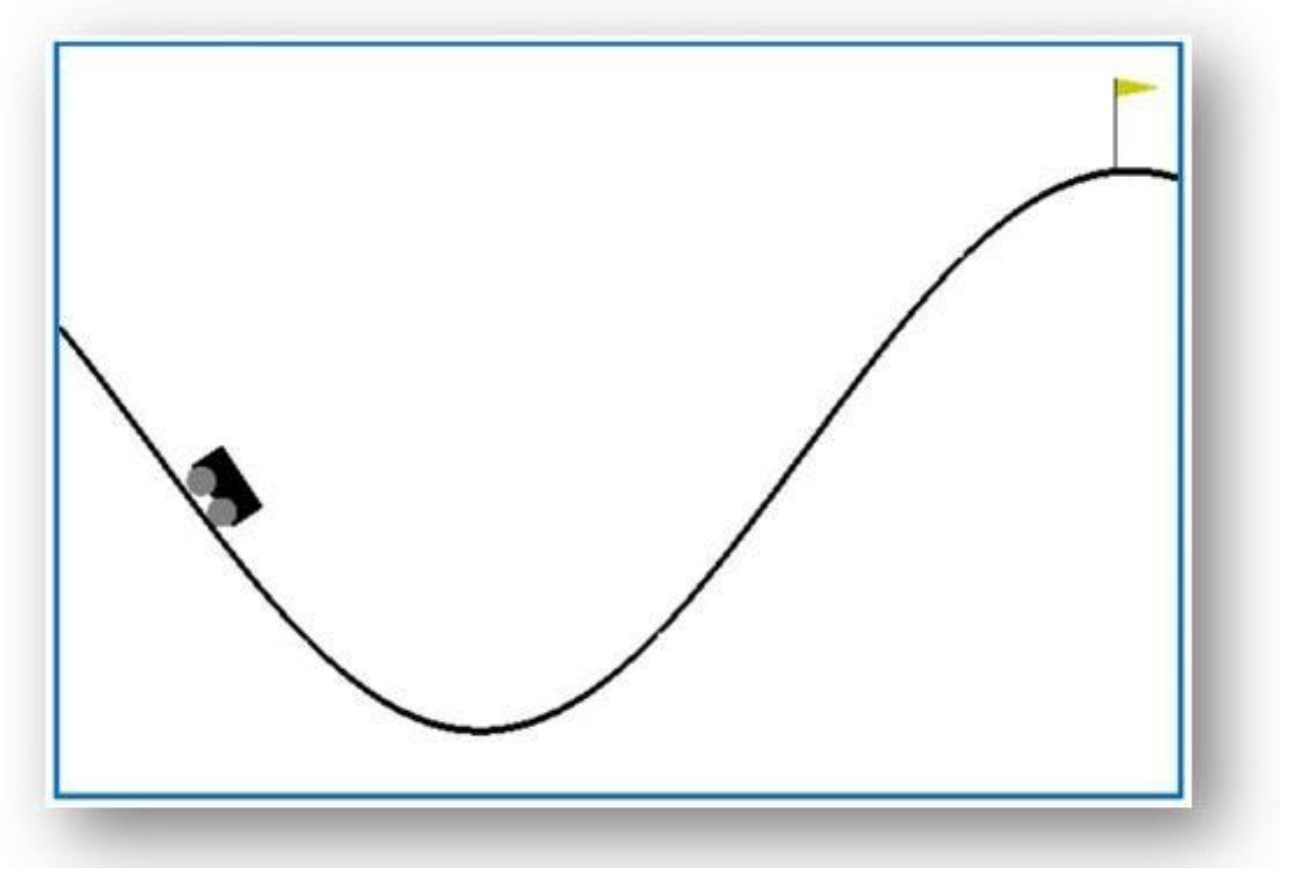

Figure 1: Mountain car environment

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

Figure 2: Action and Observation Space

The MountainCar-v0 environment is particularly challenging for reinforcement learning algorithms because of its sparse and uninformative reward signal. The agent receives a constant reward of -1 at each time step until it reaches the goal, which means there is no positive reinforcement to guide it until the task is fully completed. Complicating matters further, the agent must first move away from the goal to build momentum to climb the hill, a counterintuitive behavior that basic exploration strategies struggle to discover. Additionally, the environment has a continuous state space, making it difficult for algorithms like Q-learning to learn effective policies without function approximation or discretization. Without any intermediate rewards to indicate progress, the agent often fails to learn or converges very slowly, as it

relies on stumbling upon the goal by chance during exploration. Reward shaping addresses this problem by augmenting the reward signal, for example, by adding a small bonus based on the car's position, thereby providing informative feedback that encourages the agent to make progress toward the goal even before reaching it. This accelerates learning and helps the agent discover effective strategies more efficiently, without altering the optimal policy.

Where is the reward shaping in the example

```
###### SHAPING REWARDS #####

        shaped_reward = reward
        if done and step < 200:
          # If episode is ended the we have won the game. So, give
             some large positive reward
            shaped_reward = 250 + shaped_reward

        # Velocity is important, we give positive reward for velocity
            (is the sign correct?)
        velocity = next_obs[1]
        shaped_reward = shaped_reward + 10 * abs(velocity)

###### END SHAPING REWARDS #####
```

Listing 1: MountainCar Reward Shaping Example

In this shaping we give bonus for high velocity and we increase the end reward Try to modify this code and think what makes this agent to reach the goal.

Look at the figure and try to figure out what is the solved threshold for your agent

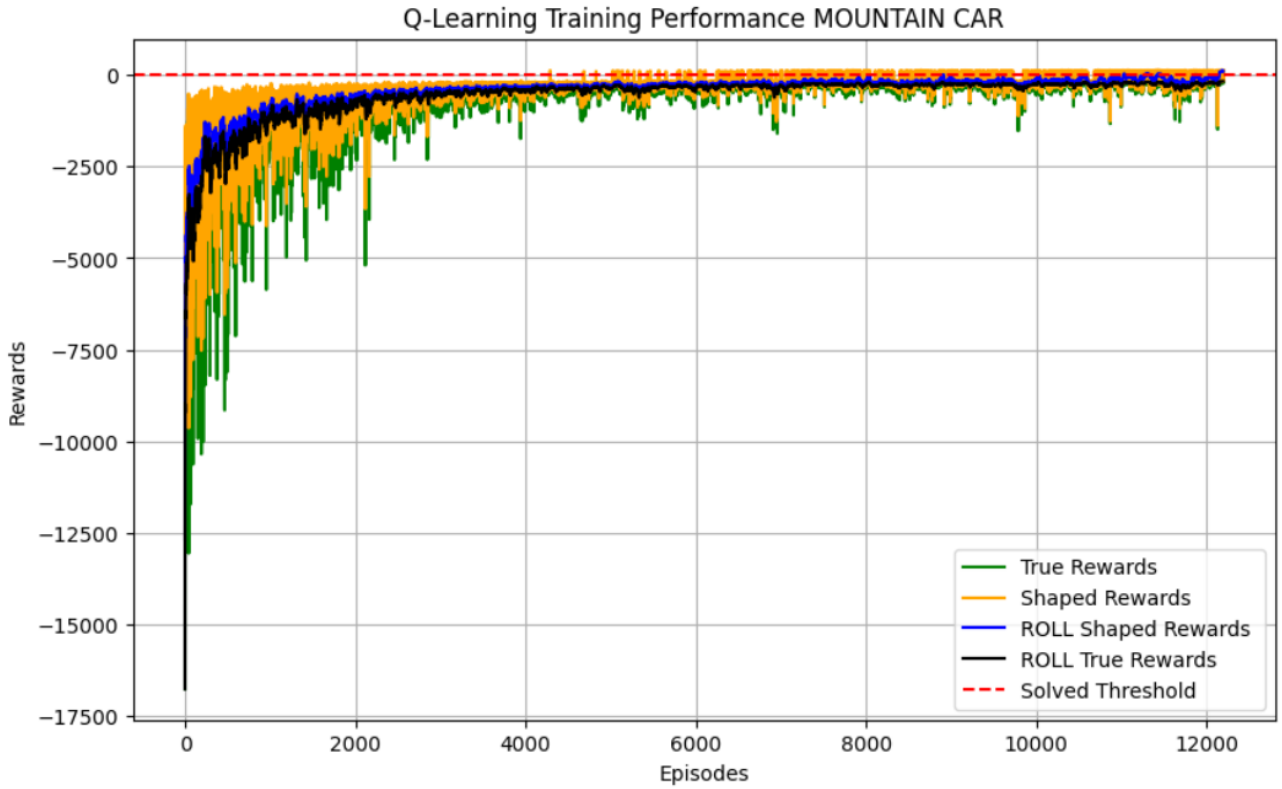

Figure 3: Learning figure

4 Deliverables and submission

Submit the code with your shaped result. HOW LONG IT TAKES TO CONVERGE?

Include your name in the title

- `Lab3_Solutions_Jose_Morales.ipynb`

Include your name in the title

References

[Far25] Farama. *Mountain Car Environment*. https://gymnasium.farama.org/environments/classic_control/mountain_car/. Accessed: 2025-06-01. 2025.