{EPITECH}
LEARN DIFFERENT*

TAXI DRIVER

ANYTIME, ANYWHERE.

* apprendre autrement

TAXI DRIVER

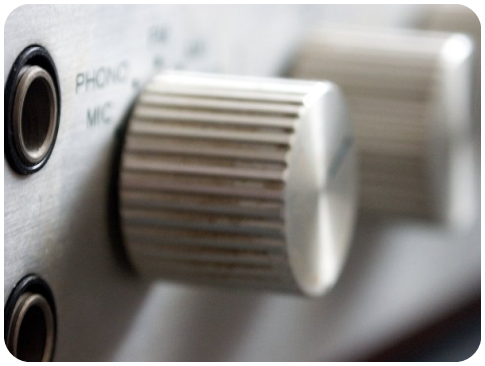
Solve the `Taxi-v3` game with an **optimized** model-free episodic learning algorithm.

The agent is a taxi that must pick up some random passengers, and drop them off at some specific locations. The game environment can be found in the *Gym* library.

You are free to choose the algorithm, whether on-policy or off-policy.
We recommand Deep Q-Learning, or Monte Carlo-based algorithms, but feel free to find the best possible approach.

If this project is not challenging enough for you, create an extension of the game where the taxi collects 2 people and obtain 4 locations for each of them, the goal being to optimize the route.

Features



Your program must include at least 2 modes :

- ✓ *user mode* allows to tune algorithm parameters ;
- ✓ *time-limited mode* uses optimized parameters to reduce the steps to solve the problem in a given time.

Both training and testing the number of episodes must be entered by the user when the program launches.

Output and delivery

Your program outputs (at least) the mean time for finishing the game and the mean rewards within all the episodes. It should also display random episodes.

To prove your algo performances, you must benchmark it against as many metrics as possible, playing around with several parameters and rewards definition.

You are also expected to develop a naive bruteforce algorithm as a comparaison point.

When a brute force algo takes ~350 steps, a fine-tuned RL model only takes 20!
The most important benchmark is the one you got with your first non-optimized algorithm. Start from here and then make improvements.

You must report these facts and figures in a document that also includes:

- ✓ commentaries about this benchmark that justify your algorithm choices and tuning ;
- ✓ your optimization strategy (concerning parameters and game rewards) ;
- ✓ different relevant graphics and arrays.

To prove the efficiency of your algorithm, you may add other comparaison algorithms.
For instance, a Q-Learning algo might be compared to a Deep Q-Learning algo.

v 1.4

* apprendre autrement