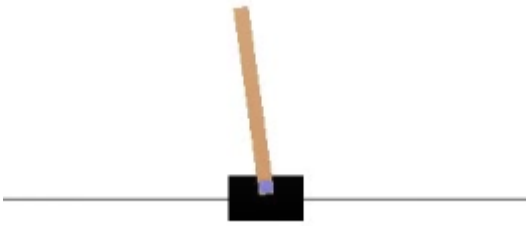
Block 4: OpenAI Gym

Applied Data Analytics and Machine Learning in Python

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

The gym library is a collection of test problems — environments — that you can use to work out your reinforcement learning algorithms. These environments have a shared interface, allowing you to write general algorithms. [Source: https://gym.openai.com]



Task 1 - CartPole-v1:

Description of the environment:

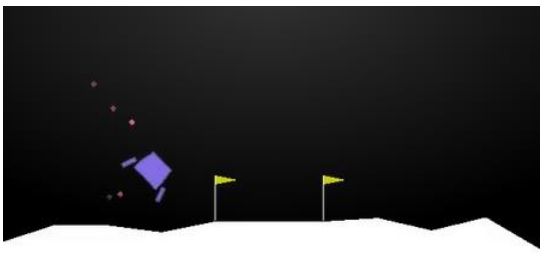
> A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force of +1 or -1 to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward of +1 is provided for every timestep that the pole remains upright. The episode ends when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center. [Source: https://gym.openai.com/envs/CartPole-v1]

Task:

> You are provided the file *CartPole_v1_QLearning_Template.py* which contains a template for a tabular QLearning algorithm to solve the CartPole_v1 problem. Complete the given functions as described in the template file.

Hints for better understanding of *discretize()*:

> The agent performs in a continuous environment but the tabular QLearning algorithm cannot work with continuous inputs. Therefore all observations must be discretized first.



Task 2 – LunarLander_v2

Description of the environment:

> In this environment a lunar lander has to be landed on a landing pad. The landing pad is always at coordinates (0,0). The coordinates are the first two

numbers in state vector. The reward for moving from the top of the screen to landing pad and zero speed is about 100..140 points. If the lander moves away from the landing pad it loses reward back. An episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. Solved is 200 points. Landing outside the landing pad is possible. Fuel is infinite, so an agent can learn to fly and then land on its first attempt. Four discrete actions available: do nothing, fire left orientation engine, fire main engine, fire right orientation engine. [Source: https://gym.openai.com/envs/LunarLander-v2]

Task:

You are provided the file *LunarLander_v2_QLearning_Template.py* which contains a template for a DeepQLearning algorithm to solve the LunarLander_v2 problem. Complete the given functions as described in the template file.