문제 1.

실습 시간에 짠 DQN 코드를 수정해서, OpenAI GYM 에서 제공하는 CubeCrash-v0 게임을 플레이 하는 Agent 를 트레이닝 하는 코드를 Python 을 이용하여 짜 봅시다.

- CubeCrash-v0 에 대한 example code 가 CubeCrash-v0.ipynb 에 포함되어 있습니다:
- 수업실습에서는 state vector 가 1 차원이였기 때문에, 이를 잘 학습할 수 있는 형태로서, build_dqn 을 2 layered fully connected layer 로 디자인 했는데, 이번에는 CubeCrash-v0 의 state vector 가 RQB 이미지로 이루어져 있는 형태 이므로 build_dqn 함수를 바꾸어서 3 layered 2D CNNs + fully connected layer 으로 바꾸어서 만들어 봅시다.
    - 3 layered-2DCNN 의 output size 는 (8,16,32), kernel size 는 3 으로 정합니다.
    - build_dqn 함수를 만들기 위해 다음의 tf.keras.layers 의 다음 함수를 이용합니다.
        - Conv2D: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D
        - Flatten: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Flatten
        - Dense: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense

- 수업에서 다루었던 dqn.ipynb 코드들을 dqn.py 파일 안에 복사해 넣고, dqn.py 파일을 main.py 에서 import 하여, dqn 모델을 training 해봅시다.
  따라서 최종적으로 만들어야 하는 파일은,
    - DQN algorithm 이 들어있는 dqn.py
    - Training 을 할 수 있는 main.py

  입니다.

문제 2.

stable_baselines3 module 에서 제공하는 PPO library 를 이용하여 CarRacing-v0 게임을 플레이 하는 Agent 를 트레이닝 하는 코드를 jupyter lab 을 이용하여 짜 봅시다.

- CarRacing-v0 에 대한 example code 가 CarRacing-v0.ipynb 에 포함되어 있습니다: