# Instructions on how my project can be used

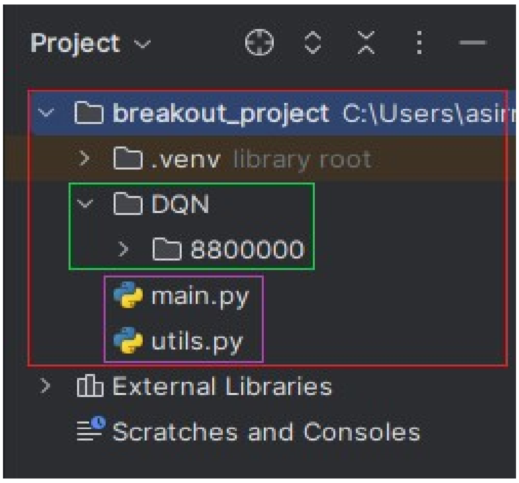
First, please take into account the requirements listed below to create an appropriate Python environment.

Python Version: 3.10.11

Required Libraries

- pip install gym==0.22.0
- pip install gym[atari]==0.22.0
- pip install gym[accept-rom-license]
- pip install opencv-python
- pip install psutil
- pip install tensorflow==2.13.0
- pip install matplotlib==3.8.2

Then, from Source Code folder, copy “main.py” and “utils.py” to working directory. Also, from Model folder, copy DQN folder to working directory so that the structure would be like following:



```
Project
breakout_project C:\Users\asir
  .venv library root
  DQN
    8800000
  main.py
  utils.py
External Libraries
Scratches and Consoles
```

My working directory name is “breakout_project”

breakout_project\DQN

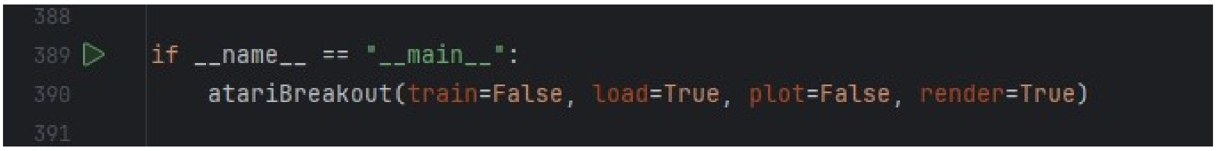
breakout_project\DQN\8800000

breakout_project\main.py

breakout_project\utils.py

After that, open “main.py” and find the following lines of code at the end of the file.

Simply, set the order as follows if you want to use a pre-trained model and see how AI plays the Atari Breakout game

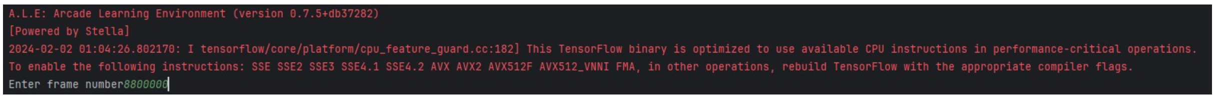


```python

if __name__ == "__main__":
    atariBreakout(train=False, load=True, plot=False, render=True)

```

Run the “main.py” file after you set these parameters. The interpreter asks you to enter input, in this case you need to enter 8800000 as “frame number”.



```
A.L.E: Arcade Learning Environment (version 0.7.5+db37282)
[Powered by Stella]
2024-02-02 01:04:26.802170: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Enter frame number8800000
```

Check the DQN folder to get that number. The “frame number” can be changed after the training. Before loading a model, be careful regarding the “frame number”. You can enjoy how AI plays the game after you enter your input.

## MORE ABOUT PARAMETER SETTING

Let’s learn more about what these parameters mean:

“train” refers to training the AI,

“load” refers to loading the pre-trained model and game data

“plot” refers to plotting the graph according the game data

“render” refers to visualization of the game.

All the combinations and what they mean are shown below:

**(train=False, load=False, plot=False, render=False)** = nothing happens

**(train=False, load=False, plot=False, render=True)** = white screen render then exit

**(train=False, load=False, plot=True, render=False)** = blank plot

**(train=False, load=False, plot=True, render=True)** = blank plot and white screen render then exit

**(train=False, load=True, plot=False, render=False)** = just load model

**(train=False, load=True, plot=False, render=True)** = load model and see how AI plays the game by rendering

**(train=False, load=True, plot=True, render=False)** = load model and plot the graph according to the loaded data, no playing

**(train=False, load=True, plot=True, render=True)** = load model, see how AI plays the game by rendering and plot the graph according to the loaded data

**(train=True, load=False, plot=False, render=False)** = train from scratch

**(train=True, load=False, plot=False, render=True**) = train from scratch by rendering

**(train=True, load=False, plot=True, render=False)** = train from scratch and after training is done, plot the graph according to trained data

**(train=True, load=False, plot=True, render=True)** = rain from scratch by rendering and after training is done, plot the graph according to trained data

**(train=True, load=True, plot=False, render=False)** = train from pre-trained model

**(train=True, load=True, plot=False, render=True)** = train from pre-trained model by rendering

**(train=True, load=True, plot=True, render=False)** = train from pre-trained model and after training is done, plot the graph according to trained data

**(train=True, load=True, plot=True, render=True)** = train from pre-trained model by rendering and after training is done, plot the graph according to trained data