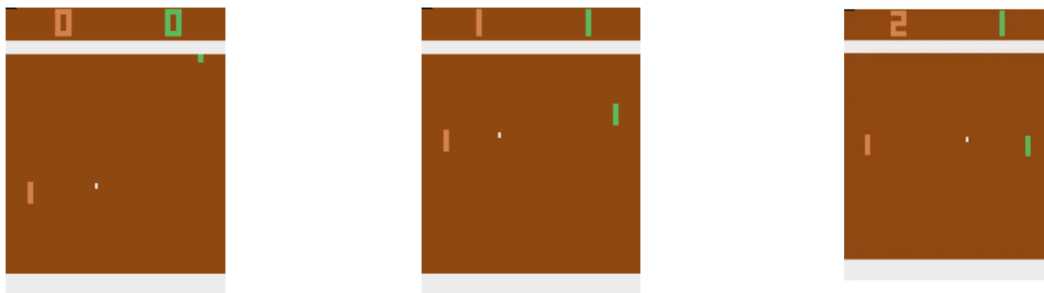


Homework 1: "Pong" Environment

Description

You control the right paddle, you compete against the left paddle controlled by the computer. You each try to keep deflecting the ball away from your goal and into your opponent's goal. Detailed documentation can be found on [the AtariAge page](#)



Action Space

Pong has the action space of `Discrete(6)` with the table below listing the meaning of each action's meanings. To enable all 18 possible actions that can be performed on an Atari 2600, specify `full_action_space=True` during initialization or by passing `full_action_space=True` to `gymnasium.make`.

Value	Meaning	Value	Meaning	Value	Meaning
0	NOOP	1	FIRE	2	RIGHT
3	LEFT	4	RIGHTFIRE	5	LEFTFIRE

For this environment, you only need to use NOOP, LEFT, RIGHT.

Observation Space

Atari environments have three possible observation types:

- `obs_type="rgb"` → `observation_space=Box(0, 255, (210, 160, 3), np.uint8)`
- `obs_type="ram"` → `observation_space=Box(0, 255, (128,), np.uint8)`
- `obs_type="grayscale"` → `Box(0, 255, (210, 160), np.uint8)`, a grayscale version of the "rgb" type

See variants section for the type of observation used by each environment id by default.

Rewards

You get score points for getting the ball to pass the opponent's paddle. You lose points if the ball passes your paddle. For a more detailed documentation, see [the AtariAge page](#).

- **+1**: When the agent successfully scores a point (the opponent fails to hit the ball).
- **1**: When the opponent scores a point (the agent fails to hit the ball).
- **0**: In all other cases (e.g., the rally is still ongoing).

Starting Sample

This is just a demo, please feel free to get creative within the required boundaries:

```
import gymnasium as gym
import numpy as np
import ale_py

gym.register_envs(ale_py)
env = gym.make("ALE/Pong-v5", render_mode="human")

obs = env.reset()
total_reward = 0

for _ in range(1000):
    env.render()

    action = env.action_space.sample()
    obs, reward, done, truncated, info = env.step(action)

    total_reward += reward
    print(f"Action: {action}, Reward: {reward}, Done: {done}")

    if done:
        obs = env.reset()
```

Task 1 [90%]

Implement **DQN** algorithm in "Pong" Environment.

Task 2 [10%]

Implement **DDQN** algorithm in "Pong" Environment.

Submission

You can use either gym [old version] or gymnasium [new version].

Draw a reward plot.

Screen record or Video record the results by rendering for at least **90 seconds**.

Pack your report with all requirements above in zip file and submit to Blackboard before March 26, 23:59pm.

Homework1_ID_Name.zip → task1 → Code.py [code file]

| Plot.jpg [image file]

| Video.mp4 [video file]

→ task2 → Code.py [code file]

| Plot.jpg [image file]

| Video.mp4 [video file]

→ report.pdf [report file]