

112-1 (Fall 2023) Semester

Reinforcement Learning

Assignment #3-2

TA: Chen-Yu Lin (林辰宇)

TA: Wei-Hsu Lee (李威緒)

Department of Electrical Engineering
National Taiwan University

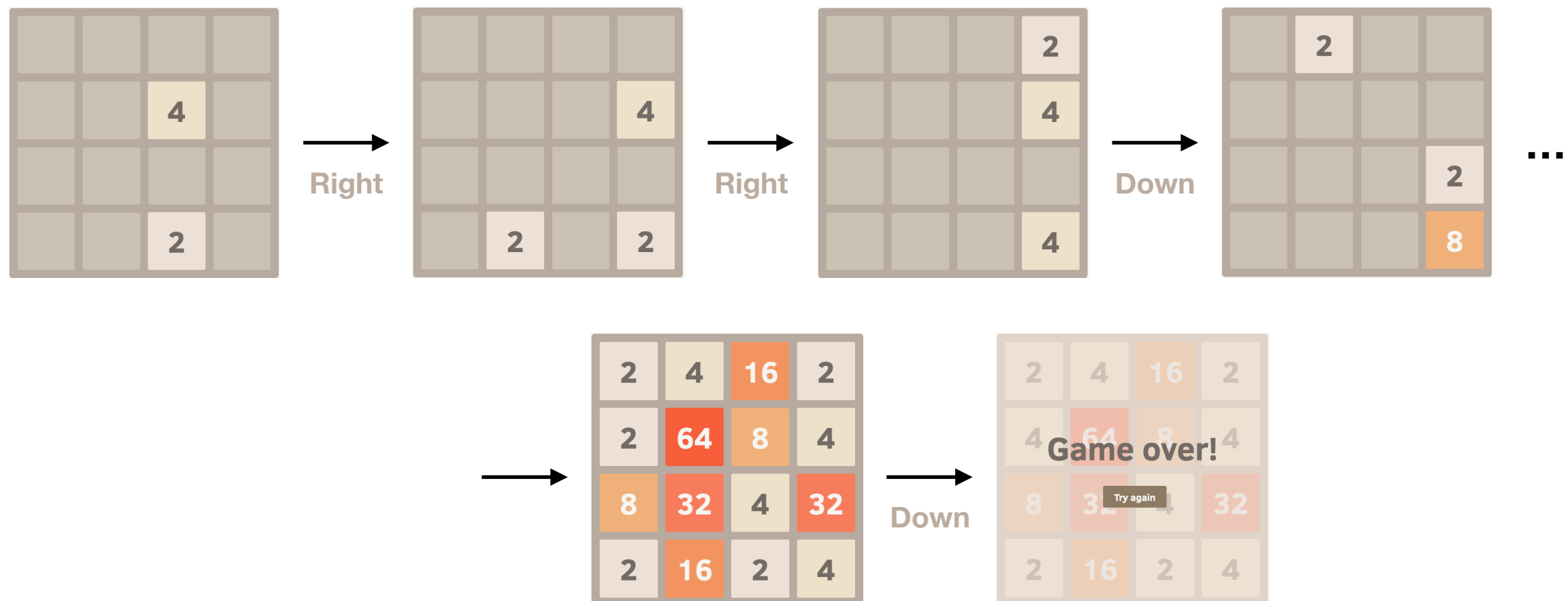
Outline

- Tasks
- Environment
- Code Structure
- Tips and Hints
- Report
- Grading
- Submission
- Policy
- Contact

Tasks

2048

- Use \uparrow , \downarrow , \rightarrow and \leftarrow to move number tiles on the board.
- When the same number collide, they will merge into a tile with the total value of two tiles.
- You will also get scores for each tiles combined.
- A new tile (with value 2 or 4) will generate in a random empty space after each move.
- The game ends when there is no legal move left.
- [Play 2048](#)



Tasks

- For this part of the assignment, we will train an agent to play 2048 with the Stable Baselines 3 package.
- A simulated 2048 environment will be provided, you will:
 - Choose and tune hyper parameters with algorithms from SB3.
 - Modify the environment (Terminal conditions, reward shaping...)
 - Work on other techniques that can help agent perform better.

Environment

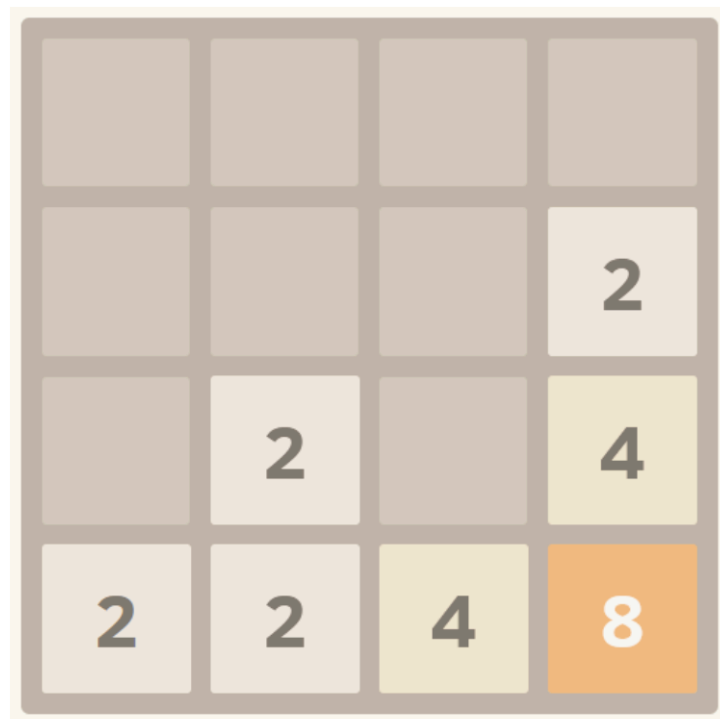
Terminal states

- For this environment, there will be 2 scenarios causing the step function to return True for termination.
 1. Termination for game end:
 - The episode terminates when there is no legal move left.
 2. Termination for illegal moves:
 - To avoid infinite loops, the environment will consider action that causes no state change an “illegal move”.
 - The episode terminates when the agent executes an illegal move.

State and Action Space

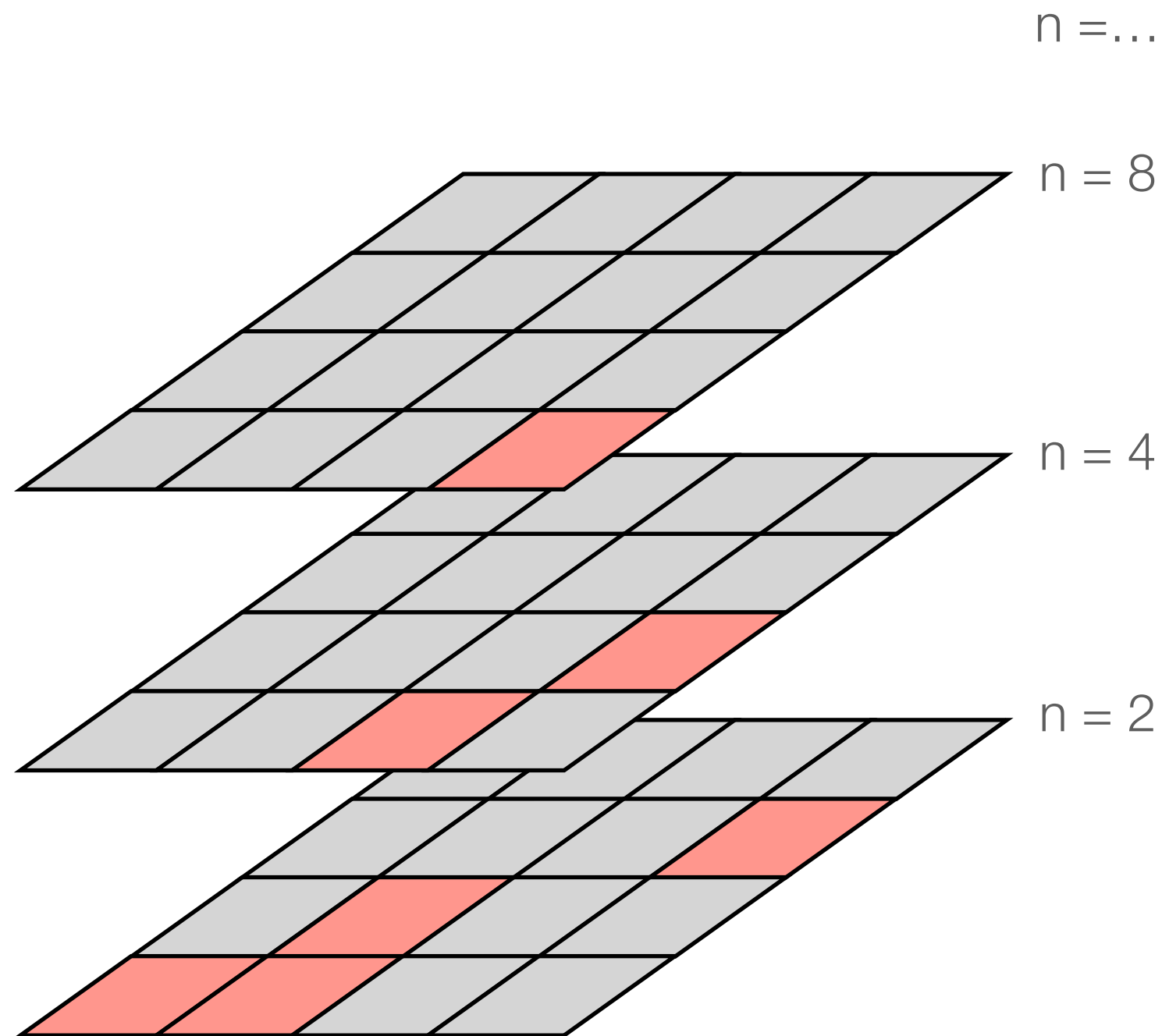
- State: Box(0,1, (16, 4, 4))
- Action: Discrete(4)

Example:



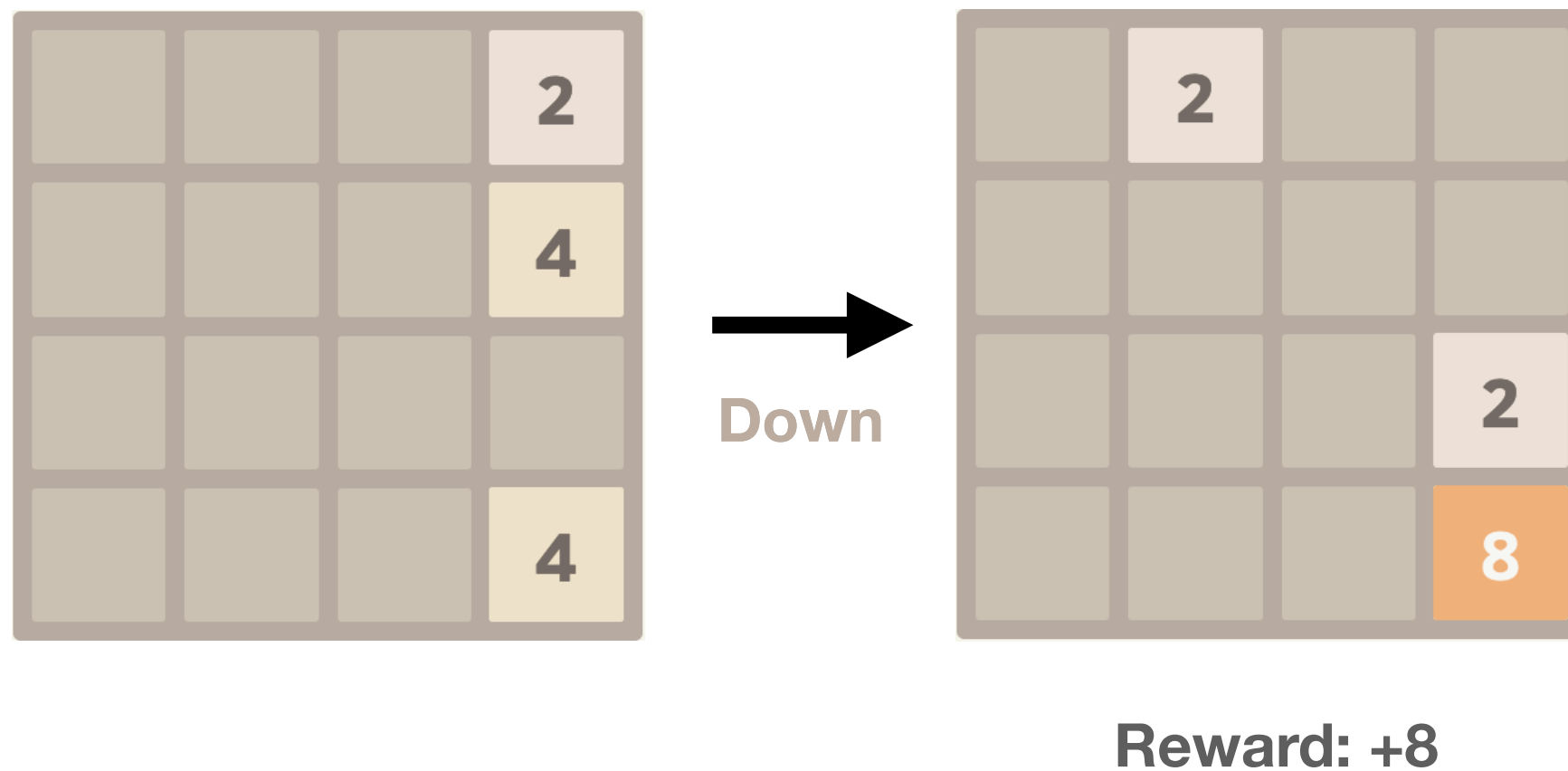
self.Matrix
(np.array)

```
[[0 0 0 0]  
 [0 0 0 2]  
 [0 2 0 4]  
 [2 2 4 8]]
```



Reward

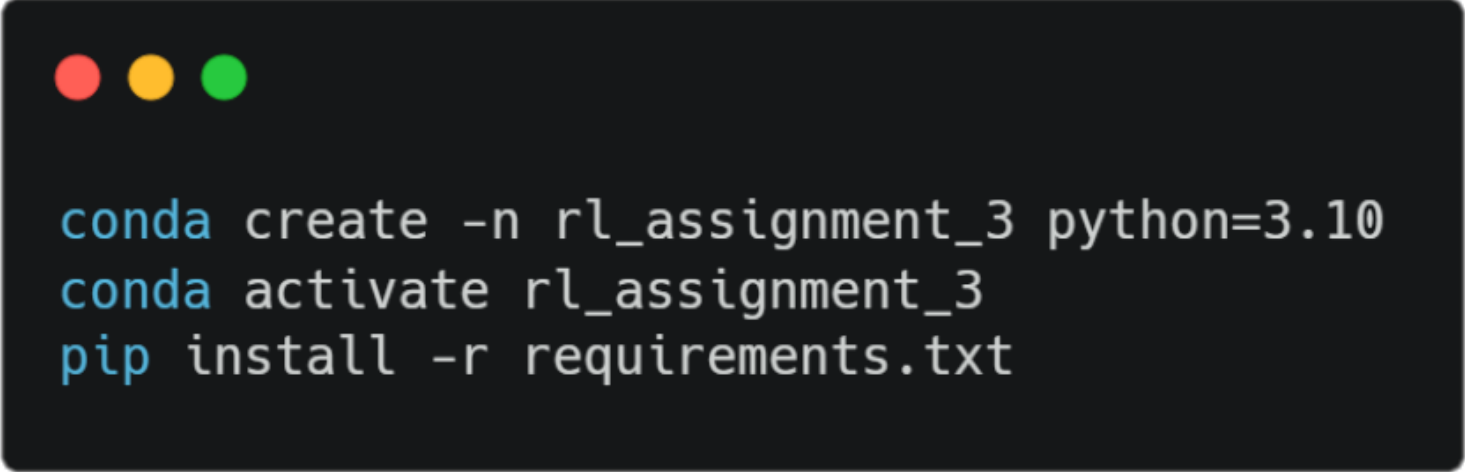
- The default reward will be the added score for each step.



Code Structure

requirement.txt

- [Conda](#)



```
conda create -n rl_assignment_3 python=3.10
conda activate rl_assignment_3
pip install -r requirements.txt
```

- [venv](#)

- Sorry for virtual environment users. Stable Baseline 3 is hard to install using venv with pip. Do not use [venv](#) in PA3.



Python Files

- [train.py](#)
 - Sample code for training your model with SB3.
 - Modify my_config to set hyper parameters and configurations.
- [my2048_env.py](#)
 - Environment used for training your agent.
 - Feel free to modify any parts of this file to help agent learn better.
- [eval.py](#)
 - Load saved model and run 100 rollouts for evaluation.
- [eval2048_env.py](#)
 - Environment that will be used for evaluate agent's performance.
 - TA will use the same “eval.py” and “eval2048_env.py” for evaluation.
- Note: You are encouraged to modify “train.py” and “my2048_env.py” as much as you like. Still, do keep in mind that the model will be put in our setting for evaluation.

Other folders

- models/
 - Folder where sample models are saved.
- envs/
 - Folder for my2048_env.py and eval2048_env.py.

Tips and Hints

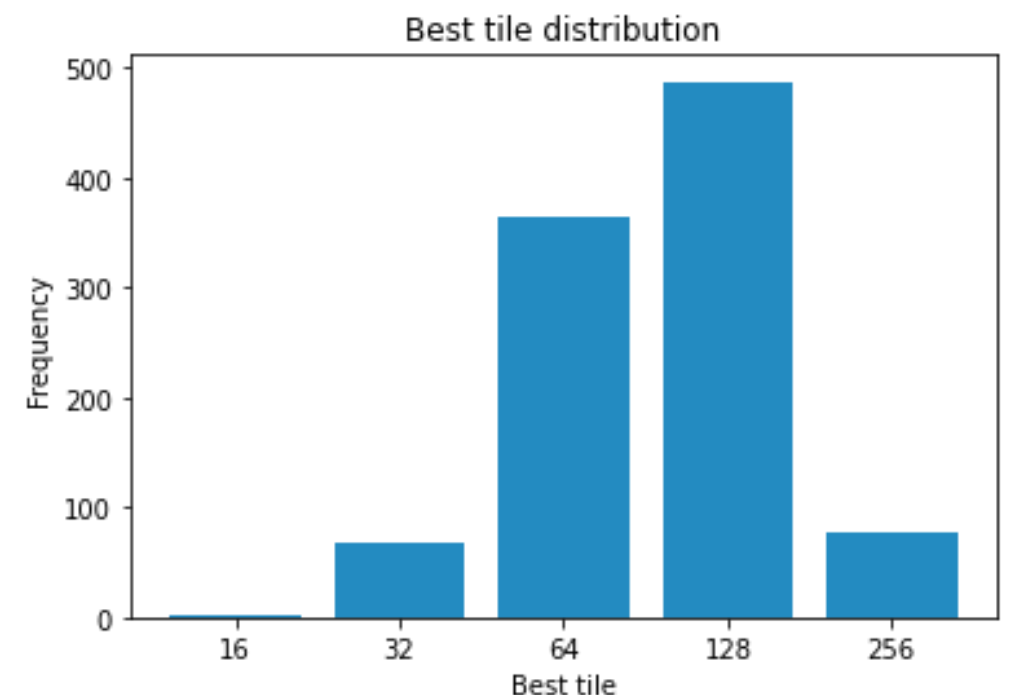
Tips and Hints

- train.py
 - Try different algorithms from SB3 and see how each of them performs. (Note that not all algorithms are compatible due to state and action space type.)
 - Try different policy networks. You can also build and use your custom network architecture and feature extractor. (See [SB3 Policy Network](#))
 - Set learning rate and/or total time step num.
- my2048_env.py
 - Set negative rewards (penalty) for illegal moves.
 - Give the agent multiple tries before terminating the episode when the agent executes an illegal move during training.
 - Set weights to give additional reward for certain board patterns that helps the agent learn a certain strategy.

Report

Report

- Q1. Discuss how you speed up your grid world environment. (10%)
- Q2. What's your best result in 2048? (5%)
 - Please show your best tile results with a histogram.
- Q3. Describe what you have done to train your best model? (15%)
- Q4. Choose an environment from the Gymnasium library and train an agent. (10%)
 - Show your results (Screenshot of environment, learning curve...)
 - Share anything you find interesting or difficult.
- LaTeX PDF format. Handwriting is forbidden.
 - [Overleaf template](#)
 - Write clear and concise in few sentences



Grading

Grading

- Baselines (30%)
 - Simple baseline: Reach 128 for at least one rollout. (Public: 5%, Private: 5%)
 - Medium baseline: Reach 256 for at least one rollout. (Public: 5%, Private: 5%)
 - Strong baseline: Reach 512 or higher for at least one rollout. (Public: 5%, Private: 5%)
- Performance (5%)
 - We will compare your total “Score” on private test cases.
 - Points will be given according to the ranking and distribution over class.
- Report (40%)
 - See “Report” slides for details


Criteria

- Test cases:
 - We will run your model on seeded environments and check the best tile value (“Highest”) for baseline evaluations.
 - Public test cases: 100 rollouts, resetting the environment with seed 0 to 99.
 - Private test cases: 100 rollouts, using another 100 seeds selected by TA.
 - Public and private will be tested separately.
 - Run time limit **3 minute** for each case to avoid infinite loops
- Only 1 model will be loaded and tested for both test cases.
- For grading, we will use “eval.py” and “eval2048_env.py” to evaluate your performance, please make sure that “eval.py” can load your model without modifying it.
- Total score on private test cases will be used for comparing your performance.
- Model should be trained with algorithms from SB3.
- Please make sure that we can briefly reproduce your results.

Submission

Submission

- Submit on NTU COOL with following **zip** file structure:
 - **gridworld.py**: Containing your implementation for assignment 3-1.
 - **train.py, my2048_env.py**: Containing your implementation for assignment 3-2.
 - We will use these files to reproduce your model.
 - **model.zip**: Saved model that can be loaded with eval.py.
 - Get rid of pycache, DS_Store, etc.
 - Student ID with lower case
 - **10%** deduction for wrong format

 **b09901171.zip**



- **Deadline 2023/11/09 Thu 09:30am**
- **No late submission is allowed**

Policy

Policy

Package

- You can use any Python standard library (e.g., heap, queue...)
- System level packages are prohibited (e.g., sys, multiprocessing, subprocess...) for security concern

Collaboration

- Discussions are encouraged
- Write your own codes

Plagiarism & cheating

- All assignment submissions will be subject to duplication checking (e.g., MOSS)
- Cheater will receive an **F** grade for this course

Grade appeal

- Assignment grades are considered finalized two weeks after release

Contact

Questions?

- General questions
 - Use channel **#assignment 3** in slack as first option
 - Reply in thread to avoid spamming other people
- Personal questions
 - DM us on Slack: **TA 李威緒 Wei-Hsu Lee**

TA 林辰宇 Chen-Yu Lin



TA 李威緒 Wei-Hsu Lee



TA 林辰宇 Chen-Yu Lin