Theory of Computer Games 2022 – Project 2+ (Bonus)

Overview: **Improve the performance of the *Threes!* player**.

1. Train a player with advanced TD methods, such as MS-TD, TC, TD(λ), OTD.
2. Integrate the TD value function with an expectimax search.
3. Use special n-tuple networks to improve the final performance.
4. Try any other improvements.

Specification:

1. The environment and the rules are the same as those in Project 1 and 2.
2. The player **should be trained by an advanced TD method**.
3. The **size of the n-tuple network is unlimited**.
4. The player could **take actions based on any improved heuristics.**
5. The program speed should be at least 100 actions per second.
6. The requirements of statistics are the same as those in Project 2.
7. Other requirements of implementation are the same as those in Project 2.

Scoring Criteria:

1. **Performance**: The minimum requirement to get the bonus is $\mathrm{WinRate}_{384} = 100\%$.
   a. $\mathrm{WinRate}_{384}$ is the reaching rate of the 384-tile in 1000 games.
2. **Report**: Summarize the improvements, the training process, and so on.
3. **The final bonus will be determined by the ranking of your work among other students.**

Submission:

1. The submission **should be archived as a ZIP file** and **named `ID.zip`**, where **`ID`** is your student ID, e.g., `0356168.zip`.
   a. Pack the **source files**, **makefile**, **report**, and other required files.
   b. Submit the archive **through the E3 platform**.
   c. **Do not upload the network weights to the E3 platform**.
      i. We will announce another location for placing weight files.
   d. Do not upload the version control hidden folder, e.g., the `.git` folder.
2. Version control (e.g., GitHub or Bitbucket) is required during the development.

References (**You are welcome to survey other methods by yourself**):

[1] K. Matsuzaki, "Systematic selection of N-tuple networks with consideration of interinfluence for game 2048," DOI: 10.1109/TAAI.2016.7880154.

[2] K.-H. Yeh, I-C. Wu, C.-H. Hsueh, C.-C. Chang, C.-C. Liang, and H. Chiang, "Multistage temporal difference learning for 2048-like games," DOI: 10.1109/TCIAIG.2016.2593710.

[3] W. Jaśkowski, "Mastering 2048 with delayed temporal coherence learning, multistage weight promotion, redundant encoding and carousel shaping," DOI: 10.1109/TCIAIG.2017.2651887.

[4] H. Guei, L.-P. Chen, and I-C. Wu, "Optimistic Temporal Difference Learning for 2048," DOI: 10.1109/TG.2021.3109887.