
Supplementary Information for In-class Game Tournament

* Issue date: 2022.04.18.

I. Tournament Plan

- **First Week (Week 10)**
 - **Theme: “Warm up”**
 - a. There will be only “preliminary games” in the first week..
- **Second Week (Week 11)**
 - **Theme: “Try something else”**
 - a. Your choice of reinforcement learning algorithm should be different from what you used in the first week.
- **Third Week (Week 12)**
 - **Theme: “Custom Policy Network”**
 - a. You should use a custom policy network architecture (i.e., your network architecture must be different from “MlpPolicy”)
 - b. https://stable-baselines3.readthedocs.io/en/master/guide/custom_policy.html

II. Grading

- **Your score will be calculated as a weighted sum of your “preliminary game score” and “tournament bonus score.”**
 - $SCORE_{total} = \sum_{week} w_i(SCORE_{pre} + SCORE_{tour})$
 - $w_1 = 1$ (Week I), $w_2 = 2$ (Week II), $w_3 = 3$ (Week III),
- First, your agent’s performance will be tested against three agents: a random agent, a poorly trained agent, the default agent embedded in SlimeVolleyGym. We will run 10 episodes for each match.
- The average scores from the matches will be converted to your league score as follows.
 - $SCORE_{pre} = \mu(scores) + 5$: this value will be between 0 and 10.
- Four teams with the highest preliminary scores will proceed to the final and the winner will have a match with Team TA.
- The teams will get the bonus points as follows.
 - First place: +4 pt (+5 if you beat Team TA)
 - Second place: +3 pt
 - Third place: +2 pt
 - Fourth place: +1 pt

III. Agent submission

- Before submission, you MUST test the compatibility of your agent with the following test code. If your agent fails to pass the compatibility test, zero point will be given.
 - EE488_04E_Volleyball_tournament_check.ipynb
- You should submit **both your agent file (zip) and the test file (ipynb)**. Your ipynb file MUST include the test logs including the game play videos.

IV. Tips

- Do not expect the following commands to magically give you what you need.
 - `model = DQN('MlpPolicy', env, verbose=1)`
 - `model.learn(total_timesteps=int(2e5))`
- Reading these pages carefully will help you successfully train an agent.
 - <https://stable-baselines3.readthedocs.io/en/master/>
 - <https://stable-baselines3.readthedocs.io/en/master/downloads/en/master/pdf/>
 - <https://github.com/hardmaru/slimevolleygym>
- Consider using Tensorboard to track the training process.
 - <https://stable-baselines3.readthedocs.io/en/master/guide/tensorboard.html>
- Check the meaning of each hyperparameter.
- Depending on the size of your network, you may not get much benefit from using GPUs. Consider using non-GPU Colab runtime which is likely to give you more computation time.
- We advise you to NOT rely only the Colab Pro Plus account especially because there will be 2~3 students in each team (and you have only one Colab Pro Plus account for each team). Try multiple things using your own Colab account and use the Colab Pro Plus account for a longer training.
- An agent that is strong against the default agent may not be necessarily strong against other agents.

V. Acknowledgement

- The skeleton code was prepared by our TAs Seungjae Han and Changyeop Shin, grad students in NICA Lab.