ASSIGNMENT 2: PART 1 [30 MARKS]
DEEP REINFORCEMENT LEARNING ON ROBOTIC AGENTS

This is a group assignment. Submit Part 1 before **October 13, 2023, noon** Phoenix time.

1 GROUP FORMATION

This is a group assignment. There must be **five** (cannot be less than five) members per group. These groups can be the same or different from the project groups. Please form groups by yourselves and enter your group formation in Sheet 1 of this spreadsheet. Note that this spreadsheet is different to what we gave for the project. Column C is only relevant to Part 1 of the assignment.

2 TASK

Study the Reinforcement Learning (RL) algorithm assigned to your group in Sheet 1. Create a 5 minute Google slides presentation on the algorithm including the following,

1. How does the RL algorithm assigned to you work? **[∼3 min, 20 marks]**
2. Pros and cons compared to some other basic RL algorithms. **[∼1 min, 4 marks]**
3. What are the applications? You might mention various properties and requirements (e.g., model-based/model-free, on-policy/off-policy, discrete/continuous state/action spaces, etc.). **[∼1 min, 3 marks]**

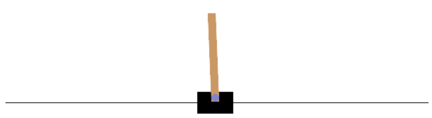
One or more group members will present during the class on 10/13/2023. It is up to the group to decide how many members will present. The presentation will be followed by a group Q&A session **[∼3 marks]**. All groups should submit the presentation slides(the file format should be *Assignment2_group_number.pptx* or *Assignment2_group_number.pdf*) before **noon 10/13/2023** via Canvas and append them here before noon October 13, 2023.

ASSIGNMENT 2: PART 2 [70 MARKS]
DEEP REINFORCEMENT LEARNING ON ROBOTIC AGENTS

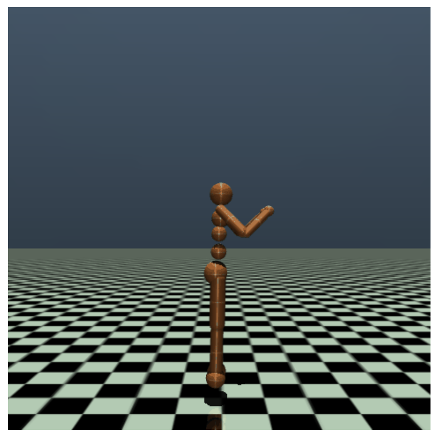
This is a group assignment. Complete Part 2 with the same group members as in Part 1 of this assignment (i.e., 5 members per group). Submit before **October 15, 2023, 11.59 pm** Phoenix time.

1 PROBLEM

For this assignment, we have two interesting robotic agents in a Gym environment to work with. Your goal is to use Reinforcement Learning (RL) algorithms to train your agent to perform a certain task. This might involve using existing libraries, reusing models, fine-tuning models, etc. Justify all assumptions used and add references to any reused codes and/or papers in the assignment report.



Cartpole env



Humanoid env

Table 1: Gym environments

2 TASK

2.1 THE REINFORCEMENT LEARNING TASK

We consider two simulated environments—the Cartpole and Humanoid for Gymnasium. Each environment has its own goal. Your goal is training the *Cartpole agent to balance itself* and the *Humanoid agent to learn to walk*. Your agent must fulfil the task for the given env.

2.2 SOFTWARE PACKAGES

Students will need to get familiarized with python packages such as Gymnasium, MuJoCo, and Stable Baselines 3. There are tutorials on how to train agents using these environments on their respective github repositories in detail. Though not mandatory, students are encouraged to use Google Colab for completing the assignment.

You are given a **Starter Code** that allows for rendering the *Cartpole* and *Humanoid* environments in a Colab space. This notebook has minimal code only for rendering with a random policy. You will need to use a T4 GPU on Google Colab for MuJoCo to work. In this section of the assignment, you are expected to run the Starter code without any errors. **[10 Marks]**

2.3 TRAINING

Use the two Gymnasium environments we discussed. There are no restrictions other than using the standard Gymnasium env **without** modifying any default env parameters. Pick a few RL algorithms. You are allowed to use any RL algorithm you see a good fit. You can start by searching **here** as well as read recent research papers. It is recommended to try multiple algorithms (at least two). Note that an RL algorithm with a particular neural network architecture that works well for the Cartpole env might not work well for the Humanoid env. Using Stable Baselines 3, obtain policies from the RL algorithms. You might need to change the neural network architectures in Stable Baselines 3. For each env, plot *reward vs episode* for all algorithms you tried in the same figure with the *best performing algorithm* in red. Add a legend to the plot. Axes and their values in the plot should be clear. You will be graded for training the agents and presenting results in a clear way.

[30 Marks (15 each env)]

2.4 RENDERING

Render the video for *best performing algorithm*. You can use the rendering function provided in the Starter code. **[20 Marks (10 each env)]**

3 SUBMISSIONS

Your Canvas submission must contain a **single .zip file** with:

1. an *.ipynb file with containing all results and video. Please finish the assignment in an *.ipynb notebook (downloadable from Google Colab). The final notebook **must** contain the entire source code, reward plots, and a rendered video with policies for all RL algorithms you tried. **To reiterate, the graders should be able to see final results (plots, text, etc.) as soon as they open the notebook, before re-running the code.**
2. two *.png images (one for each env) of reward vs. episode curves of the *best performing policy* working on agents. Mark the best performing policy in red.
3. two *.mp4 videos (one for each env) of the *best performing policy* working on agents. Do not submit videos for poorly performing policies. The video can be downloaded from Colab.
4. a descriptive *.pdf report containing the RL algorithms you tried, all your assumptions, results, and references. Discuss why the best performing algorithm provided the best result. The report can be up to **four pages maximum** with all figures and references included. **[10 Marks]**

4 POLICY REGARDING EXPECTED STUDENT BEHAVIOR

Students in this class are expected to acknowledge and embrace the FSE student professionalism expectation located at: `https://engineering.asu.edu/professionalism/`.

4.1 ACADEMIC INTEGRITY

Students in this class must adhere to ASU's academic integrity policy, which can be found at ASU Integrity Policy. **Students are responsible for reviewing this policy and understanding each of the areas in which academic dishonesty can occur**. All engineering students are expected to adhere to the ASU Academic Integrity Honor Code. All work submitted for the course cannot have been submitted for any other course or any previous section of this same course. Student academic integrity violations are reported to the Fulton Schools of Engineering Academic Integrity Office (AIO). Withdrawing from this course will not absolve you of responsibility for an academic integrity violation and any sanctions that are applied. The AIO maintains a record of all violations and has access to academic integrity violations committed in all other ASU college/schools.

Carefully read the Course Syllabus for late submission and other policies. If you find yourself stuck on something, start a discussion on Canvas and/or come to TA office hours.

References and Links

1. Gymnasium: https://github.com/Farama-Foundation/Gymnasium
2. Mujoco: https://github.com/google-deepmind/mujoco
3. StabeBaselines3: https://github.com/DLR-RM/stable-baselines3
4. Starter Code: https://colab.research.google.com/drive/1euZlyv0jebmxizLZfF7YiRvUXXm5kDie
5. RL Algorithms: https://stable-baselines3.readthedocs.io/en/master/guide/algos.html