**ME5418-group23 Neural Network report**

**Code explanation:**

Due to the lack of documentation for the Legged Gym and RSL-RL libraries initially chosen by our team, understanding the code proved to be quite challenging. Therefore, we made a significant adjustment this week by switching to the IsaacGymEnv and RL-Game libraries. We used the IsaacGymEnv library to create reinforcement training actors and environments, and the RL-Game library to conduct reinforcement learning experiments. The documentation for these two libraries is relatively complete, and we have selected a few key documents to place in the ref folder. Here is a summary of the main tasks completed by each team member this week:

**Isaacgymenvs/tasks/go2.py(go2\_test.py)**: This script uses the Isaac Gym RL framework to create a new task by defining a Task class that extends from VecTask (provided by the IsaacGymEnvs library). The first step is to create the \_\_init\_\_ function. Additionally, there are a few essential methods that must be implemented by any subclass of VecTask: create\_sim, pre\_physics\_step, and post\_physics\_step. Detailed descriptions of these methods and their required functionality can be found in ref/isaacgymenv/framework.md. This environment interacts with the RL algorithms in RL-Game through obs\_buf, rew\_buf, reset\_buf, and extra. The go2\_test.py script is where we are currently experimenting with various modifications to the environment, including adding complex terrains, incorporating observations of external environmental factors, improving the reward function and improving terminate function. These changes have not yet been merged into go2\_test.py, but we plan to integrate them in future development stages.

**Isaacgymenvs/cfg/task/go2.yaml**: This configuration file provides core settings for a reinforcement learning environment, including parameters for physical simulation, control, learning, and randomization. It is suitable for a GPU-based continuous control task and offers flexibility for adjusting initial states and domain randomization.

**Isaacgymenvs/cfg/train/go2PPO.yaml**: The configuration file sets all key hyperparameters for an A2C reinforcement learning training environment, suitable for GPU-accelerated continuous control tasks, with flexibility domain randomization.

**Note:** Domain randomization is a feature provided by IsaacGymEnvs that helps achieve sim-to-real transfer. We have not enabled it for two reasons: first, our project does not require sim-to-real capabilities, and second, the official apply\_randomizations function seems to be missing the definition for the get\_attr\_val\_from\_sample function, which currently prevents us from using it. And due to extremely limited time this week, most of the reinforcement learning algorithm relies on the existing RL-Game library. Currently, we can only modify certain hyperparameters through Go2PPO.yaml, and we haven’t been able to make deeper adjustments to the model structure or algorithm. Additionally, the visualization of training results is still incomplete. I will work to complete these sections next week.

**Existing Codes/Libraries:**

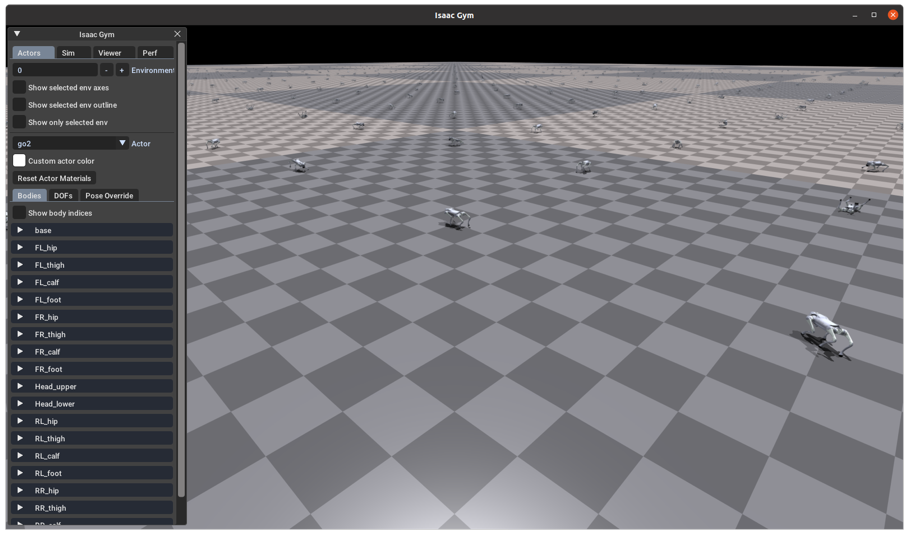
RL-game: RL-game is a Python library for developing and testing reinforcement learning environments and agents. It provides simple interfaces and tools to help users create custom RL game environments and algorithms for conducting reinforcement learning experiments. The library is designed with readability and extensibility in mind, making it suitable for rapid prototyping and experimentation.

IsaacGymEnvs: Since the RL-Game library itself does not provide an interface for the Isaac Gym environment, we incorporated the IsaacGymEnvs library. This allowed us to create environments and conduct experiments using the RL-Game reinforcement learning framework.

Hydra: The task and training configurations in the IsaacGymEnvs library are implemented using Hydra's configuration groups. To ensure consistency, we have also managed our custom task and training configurations through Hydra.

**Reflections/Lessons Learned**:

In our previous project, our team initially used the Legged Gym library along with the rsl-rl reinforcement learning framework. However, due to limited documentation provided by the developers and the lack of online resources for these libraries, we encountered significant challenges in understanding the source code within the time constraints. Consequently, we switched to using the IsaacGymEnvs library, a reinforcement learning framework developed by Nvidia adapted for the RL-Game library and the Isaac Gym environment. IsaacGymEnv and RL-Game both come with well-maintained documentation, which has greatly facilitated our secondary development process.



**Figure 1: Training Process Visualization**