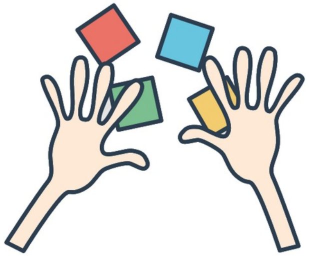

Hands-on: #1

Lab Exercise: Comparing A2C, TRPO, PPO, and DDPG

1. Run the Baseline Codes
   - Use the provided scripts for each algorithm (e.g., `trpo.py`, `ppo.py`, and `ddpg.py`, a2c,).
   - Train on the same continuous-action environment (Pendulum-v1 or LunarLanderContinuous-v2) or differentiating between environments (e.g., `CartPole-v1` for TRPO/PPO or `Pendulum-v1` for DDPG).
   - Note the final average returns or any success metrics.
   - Also make runs using StableBaseline3

2. Tune Hyperparameters
   - For each algorithm, choose one parameter to change (e.g., learning rate, batch size, clipping range, etc.).
   - Re-run the training with this modified parameter and observe whether performance improves or worsens.

3. Save and Evaluate
   - Add code to save the trained model parameters (actor and/or critic) in a file (e.g., `ppo_policy.pth`).
   - Write a small function (or code in `main`) that loads these parameters and runs several episodes without training, printing or recording the test returns.

4. Compare Results
   - For each algorithm, record the training curves (or final returns) in a simple text file or CSV.
   - Plot or compare these results side by side: Which algorithm converges faster? Which reaches the highest score?
   - Provide a brief explanation of any differences you observe.