Homework 2

Release: 02/06/2023 Due: Sun. 02/12/2023, 11:59 PM

- You are **allowed** to consult any external resources, but you must cite them. You are also **allowed** to discuss with each other, but you need to acknowledge them. However, your submission must be your own work; specifically, you must **not** share your code or proof.
- Your submission includes a PDF file containing the solutions and results.
- This homework is worth 10/100 of your final grade.
- This homework itself contains 20 points with 10 extra credits.
- It is highly recommended that you begin working on this assignment early.

Problem 1. Motion Planning (18pts).

- First, please go over the motion planning tutorial at `https://sapien.ucsd.edu/docs/latest/tutorial/motion_planning/index.html`
- Use the knowledge you have learned to write a planner+controller policy to solve the LiftCube task with state space (no need for visual observation) in the tutorial of HW1.

The following are some detailed instructions.

- You need to git clone the ManiSkill2 repo which contains the URDF and SRDF files for the robot used in the LiftCube environment, namely,

  `ManiSkill2/mani_skill2/assets/descriptions/panda_v2.urdf`

  `ManiSkill2/mani_skill2/assets/descriptions/panda_v2.srdf`

- You can refer to the following file for how to access object poses, what is the success condition, and some hints on how to solve the task (from the reward function), etc.

  `ManiSkill2/blob/main/mani_skill2/envs/pick_and_place/pick_cube.py`

- Due to a naming issue, please rename all

  `/mani_skill2/assets/descriptions/franka_description/meshes/*.stl`

  `/mani_skill2/assets/descriptions/franka_description/meshes/finger.stl.convex.stl`

- The robot object can be accessed via `env.agent.robot` where `env` is the environment instance created in ManiSkill2.
- Due to the nature of the planner and the IK solver, there might be failures during motion planning.
- You need to solve the task by providing videos of at least 5 successful trajectories.
- Please also submit your code (e.g., in a Jupyter notebook).

Problem 2. Building a Toy Agent (10pts extra credit).

- First, please go over the SAPIEN tutorial at `https://github.com/haosulab/SAPIEN-tutorial`
- Based on what you have learned in the tutorial, build a simple toy environment that has multiple articulated links and can perform simple activities. Record a video of your agent movement in .mp4 format. Submit your codes and the video as a single zip file `HW2_solution.zip`.
- You will get 5 extra credits by showing a self-built environment. You will receive another 5 extra credits if you are selected as the top 3 most creative ideas. We will show your creation in class if selected!

Problem 3. Feedback (2pts).

Please include any feedback you have on this homework.