Instruction for Project 3

Contest Week

1. Introduction

In Project 3, you will compete against other teams using records from the simulation. You are allowed to use any algorithm, such as Gaussian Process Regression and Reinforcement Learning. Similar to Project 2, you will use 720-dimensional LiDAR scan data to predict the action for your rccar. You may create new files to define classes or train models, but make sure to upload all files required for running '{TEAM_NAME}_project3.py' to GitHub. If you want to use specific Python modules, let the TA know so they can be installed on our evaluation server. If you plan to use deep learning modules, we kindly ask you to use PyTorch.

2. Large File

We recommend you not to use too large model because it may not run in real-time speed when running real world RC car. Nevertheless, if the size of the model is too large to be pushed to the git repository, please upload the model using one of the following two methods.

2.1 Git LFS

Git LFS allows you to push large model files to your git repository. If you want to use Git LFS, please follow the instructions below.

1. Install Git LFS on your PC.

```
sudo apt-get install git-lfs
```

2. You need to declare that you will be using Git LFS in your local repository. You can declare it by typing the following command in the terminal in your local repository.

```
(in your local repo) git lfs install
```

3. Add the file types to track (e.g. *.pkl), and git add .gitattributes.

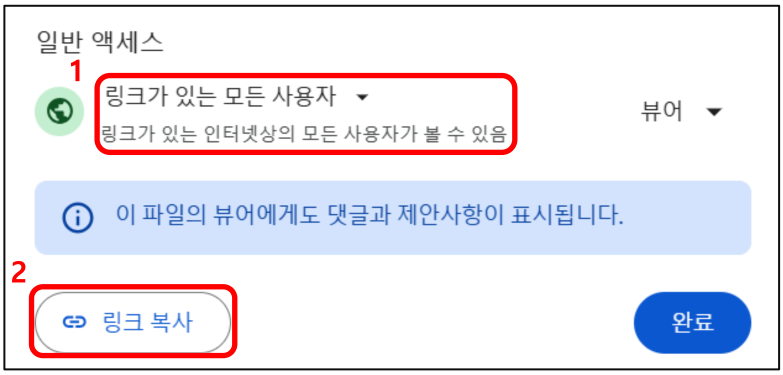
```
git lfs track "*.pkl"
git add .gitattributes
```

You can now use git as same as before.

2.2 Google Drive

If you have reached the limit of available Git LFS usage, please use Google Drive to upload your model file. If you're going to use Google Drive, your model file name and google drive file ID must be specified in 'model_link.json'. Please follow the instructions below.

1. Go to Google Drive and upload your model file.
2. Set the file to be 'Anyone with the link' and copy the link.



일반 액세스

링크가 있는 모든 사용자

링크가 있는 인터넷상의 모든 사용자가 볼 수 있음

뷰어

이 파일의 뷰어에게도 댓글과 제안사항이 표시됩니다.

3. The copied link will have a form as follows.

https://drive.google.com/file/d/**1q5mGgVg3STfRtw8VQBsv**/view?usp=sharing

Your 'FILE_ID' is the part between '/d/' and '/view', highlighted in red. Get your 'FILE_ID' from the link.

4. Inside the 'model_link.json' file in '~/project/IS_{TEAM_NAME}/project/model_link.json', replace "my_model.pkl" and "GOOGLE_DRIVE_FILE_ID" part with your model name and ID.

```
ex) "my_model.pkl": "GOOGLE_DRIVE_FILE_ID" → "our_model.pkl": "1q5mGgVg3STfRtw8VQBsv"
```

3. Setup

Don't forget to change your 'setup.py' to add project3 on ros2 package.

```
'RLLAB_project3 = rccar_bringup.project.IS_RLLAB.project.RLLAB_project3:main'
# Replace RLLAB with your team name
```

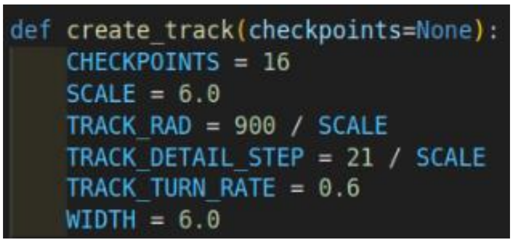
4. Random Map Generation

In our class repository (https://github.com/rllab-snu/Intelligent-Systems-2024), you can find 'random_trackgen.py' file added in 'maps' folder. By running this code, you can randomly generate maps to use for training your model.

```
cd [your-workspace-folder]/maps

# you need to use different seed to make different maps.
python random_trackgen.py --seed your_seed --name your_map_name
```

Inside the 'random_trackgen.py' file, you can change the following parameters in the "create_track" function to create more dynamic maps.



```
def create_track(checkpoints=None):
    CHECKPOINTS = 16
    SCALE = 6.0
    TRACK_RAD = 900 / SCALE
    TRACK_DETAIL_STEP = 21 / SCALE
    TRACK_TURN_RATE = 0.6
    WIDTH = 6.0
```

5. Evaluation

You can monitor your model's performance and scores on the leaderboard which is available on our website. Your algorithm will be tested on 10 unknown maps. For each map, rankings are determined based on 2 criteria in order (1. waypoint, 2. lap time), and points are awarded accordingly: 1st place gets 20 points, then 17, 14, 11, 9, 7, 5, 4, 3, 2, 1. Your team's final ranking is determined by sum of all the points across 10 maps. The leaderboard will display your team's best performance based on the highest overall score.

To evaluate your code in a local PC, follow the instructions described in the class repository (https://github.com/rllab-snu/Intelligent-Systems-2024). To evaluate your code on the server, commit and push your codes on github and make a query on our project web server. Before you add any queries to the server, please ensure that your model file has been properly uploaded either in your github repository or in your Google Drive.

Project 3 is from **12/2(Mon) 00:00 KST to 12/6(Fri) 23:59 KST**, and the late query will not be submitted.

Have fun!!