**GRAIC 2023 Installation Guide**

Yan Miao

# **GRAIC 2023 Overview:**

You will be provided with all Ground Truth Perception data(E.g waypoints, lane information, obstacle information). The task is to focus on planning + control, so that your controller can navigate the vehicle through different scenarios(E.g. car cutting in lane) in different maps. Your controller will be scored on how fast you can finish one lap.

# **Demo**

Below is Yan’s Simple Baseline Controller’s Demo Video:

<https://drive.google.com/file/d/1u4rj-sxMd6fuTnt_hS9PL_gUbfWdRjyO/view?usp=sharing>

# **Submission Files**

agent.py

requirements.txt

* Any python package you used in agent.py has to be included here
* E.g. numpy==1.23.5

# **Benchmark**

Our baseline controller should be able to finish one lap on Shanghai tracks(the default track in our code) with the Tesla Model 3(the default model set in our code) with the below score.

Score for Shanghai Track without scenarios: **130**

Score for Shanghai Track with scenarios: **155 (no collision)**

# **Dependencies:**

* Carla 0.9.13
* Scenario Runner 0.9.13
* Python3.7+

# **Install the Carla 0.9.13 Simulator**

*(\* You can find the official installation guide here:* [*https://carla.readthedocs.io/en/latest/start\_quickstart/#carla-installation*](https://carla.readthedocs.io/en/latest/start_quickstart/#carla-installation)*)*

But I’ll give you a simpler installation walkthrough:

1. Download Carla 0.9.13 simulator here: <https://carla-releases.s3.eu-west-3.amazonaws.com/Linux/CARLA_0.9.13.tar.gz>
2. Extract it into a path you favored (In my case: /home/yanmiao2/ws/CARLA\_0.9.13)
3. (ECE484 students do not need to execute this step in lab machines)
   1. *sudo apt install libomp5* per this [git issue](https://github.com/carla-simulator/carla/issues/4498)
4. Add Carla API to your own python path by appending the below lines into your bashrc file( ~/.bashrc) and source the file.
   1. *export PYTHONPATH=$PYTHONPATH:/home/yanmiao2/ws/CARLA\_0.9.13/PythonAPI/carla/*
   2. *export PYTHONPATH=$PYTHONPATH:/home/yanmiao2/ws/CARLA\_0.9.13/PythonAPI/carla/dist/carla-0.9.13-py3.7-linux-x86\_64.egg*
5. Install pygame
   1. *python3 -m pip install --user pygame numpy*

If you finish this section correct, then the below 2 steps should get you open the simulator correctly:

1. *cd /home/yanmiao2/ws/CARLA\_0.9.13/*
2. *./CarlaUE4.sh*

****

# 

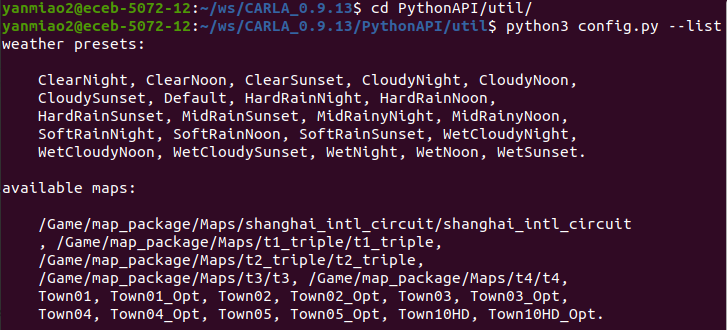
# **Install GRAIC Customized Maps for Carla 0.9.13**

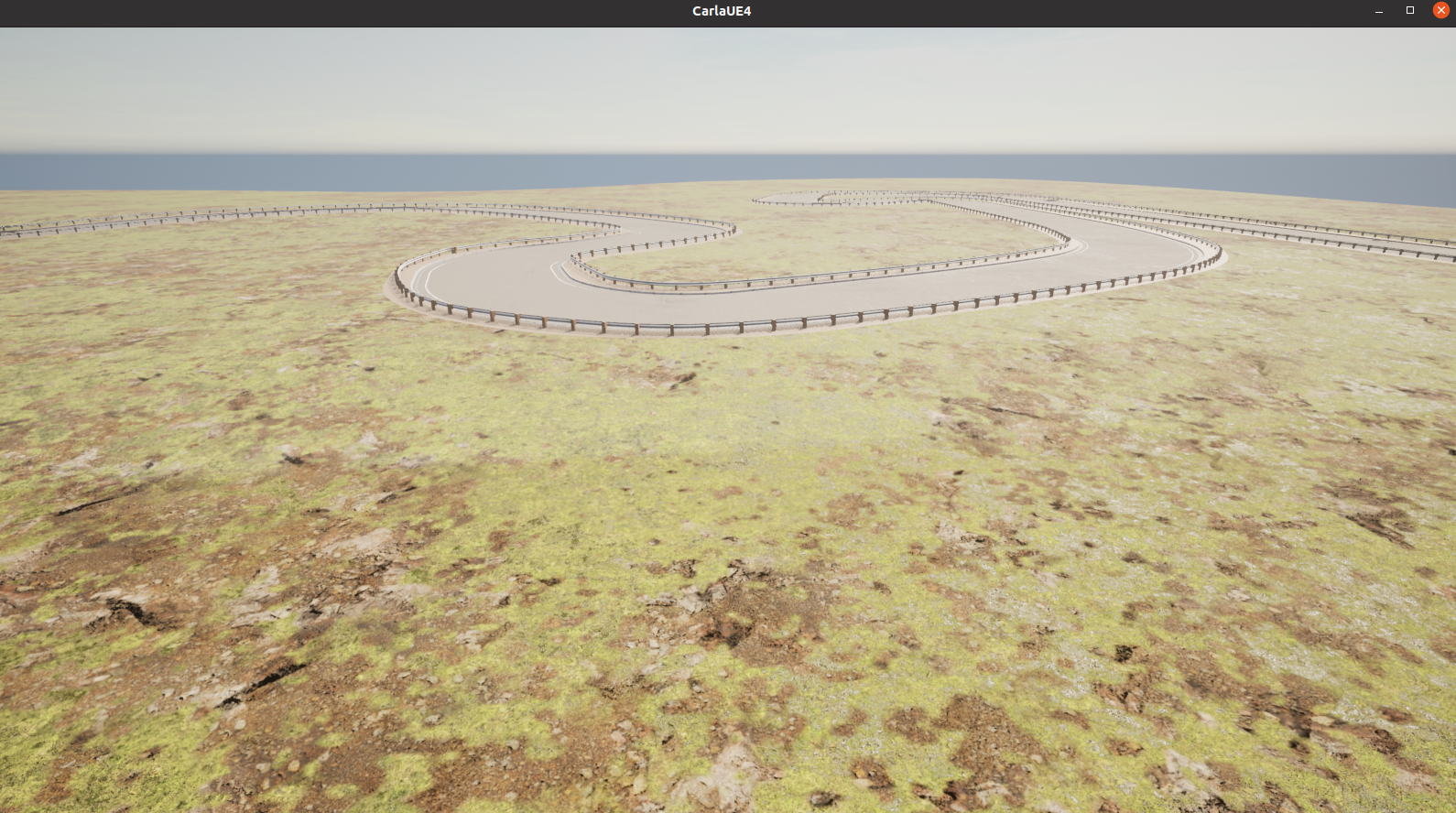
*(\* You can find the official installation guide here:* [*https://carla.readthedocs.io/en/0.9.13/tuto\_M\_add\_map\_package/*](https://carla.readthedocs.io/en/0.9.13/tuto_M_add_map_package/)*)*

But I’ll give you a simpler installation walkthrough:

1. Download the 5 raw maps in [my google drive](https://drive.google.com/drive/folders/13xQbCmlUFVmyqN-elPB8LHSIBsaeLbnZ?usp=sharing)
2. Place the 5 tar.gz file in your corresponding import path (In my case, /home/yanmiao2/ws/CARLA\_0.9.13/Import)
3. Import the map using the below 2 commands
   1. *cd /home/yanmiao2/ws/CARLA\_0.9.13*
   2. *./ImportAssets.sh*

If you finish this section correct, then the below 4 steps should get you open the simulator and switch to any of the 5 maps:

1. *cd /home/yanmiao2/ws/CARLA\_0.9.13/*
2. *./CarlaUE4.sh*
3. (new terminal) *cd /home/yanmiao2/ws/CARLA\_0.9.13/PythonAPI/util*
4. *python3 config.py --list*
   1. You should now see 5 maps’ name(Shanghai, t1, t2, t3, t4) you just imported
5. Now you can switch to any of the new map using this command
   1. *python3 config.py -m /Game/map\_package/Maps/shanghai\_intl\_circuit/shanghai\_intl\_circuit*
   2. Replace the above shanghai with t1/t2/t3/t4, then you can switch to different GRAIC maps



# **Install GRAIC Customized Scenario Runner**

*(\* We are using a customized Scenario Runner, so please do NOT pull from the official SR repo, but USE our GRAIC repo)*

1. Download the CUSTOMIZED 0.9.13 Scenario Runner from GRAIC public repo <https://github.com/PoPGRI/scenario_runner>
2. Extract SR into your workstation (in my case: /home/yanmiao2/ws/scenario\_runner-0.9.13)
3. Install SR Dependencies by these 2 lines:
   1. *cd /home/yanmiao2/ws/scenario\_runner-0.9.13*
   2. *python3 -m pip install -r requirements.txt*
4. Add SR API to your own python path by appending the below line into your bashrc file( ~/.bashrc) and source the file.
   1. *export PYTHONPATH=$PYTHONPATH:/home/yanmiao2/ws/scenario\_runner-0.9.13*

If you finish this section correctly, then executing the below scripts(E.g. copy the below 2 lines to test.py) in any path with python3(E.g. python3 test.py) should not get you any error message

1. *from scenario\_runner import ScenarioRunner*
2. *print("Success")*

# **Install GRAIC Customized Scripts**

1. Download GRAIC infrastructure scripts for 2022 <https://github.com/PoPGRI/Race/tree/main>
2. Place the folder into your workspace (In my case: /home/yanmiao2/ws/Race)
3. **Implement your controller in agent.py, where all your inputs are the Ground Truth Perception Data, and you just need to implement the controller.**

Now you are ready to Launch the whole GRAIC:

1. (Terminal 1) Launch Simulator -- check command above
2. (Terminal 2) Change to Shanghai Map -- check command above
3. (Terminal 3) *python3 wrapper.py* (in the ws/Race folder)
   * 1. Note: you may use pkill to kill python if you car dies and the scenario is kept running
     2. Check the code in the wrapper.py to enable/disable the scenarios to help you debug

Note:

1. **Only File to Modify is agent.py, we will overwrite other files when grading:**

Agent.py -- Please check the detailed comments there

1. **Carla Python API**:

You might want to become best friends with this buddy: <https://carla.readthedocs.io/en/0.9.13/python_api/>

1. **Model + Track**:

The code we release to you only runs Tesla Model 3 on the Shanghai Track, but during testing time we may use different models + different tracks.

1. **Scoring & Collision**

Currently the score of your controller for each race is defined as time to finish a lap on each race. We will first rank controllers by number of collisions; if the number of collisions are the same, we will rank you score(time to finish)

And to make life easier for you guys, your racing score will not be directly penalized by collisions, however note that:

* Collision might lead the car into “interesting” states(E.g. Flipping upside down), as defined by the carla’s physics
* Collision might break your controller, if your controller is using some neural network or caching previous history states (due to significant discontinuities)