2024 ACC Self-Driving Car

Student Competition Handbook

At their algorithmic core, self-driving cars excel or are held back in their capacity for autonomy by their knowledge and understanding of the state of the environment surrounding the car. This, debatably, makes accurate sensor fusion and state estimation the most crucial element of a self-driving stack.

The state estimation challenge has been approached by various companies using a wide variety of methods, from different sensor suites to the fusion of real-time sensor data with mapping and navigational databases. How that data is gathered, processed, and leveraged represents the grand challenge of self-driving and the motivation for the ACC Self-Driving Competition.

Competition Outline

Welcome to 2024 ACC Self-Driving Car Student Competition, powered by Quanser!

If you are a returning competition group, welcome back!

You are in the role of an autonomous driving engineer and have been tasked with the following challenge:
Complete the racing line shown in Figure 1.

Figure 1: Top-down view of virtual world

Goal: Complete the racing line shown in Figure 1. The competition is divided into 3 stages:

- **Stage 1**: Virtual Design and Submission

  Note: At the end of stage 1, registered teams will be ranked, and a set of finalists will be announced to advance from stage 1 to stages 2 and 3.

- **Stage 2:** Algorithm Validation on Physical Vehicle

- **Stage 3:** On-site Demonstration and Competition

Stage 1: Virtual Design and Submission

As a registered student team, you automatically participate in Stage 1. This is a qualifying stage, please submit your best result!

In general, teams will be ranked using the following criteria:

1. Successful completion of specified circuit.
2. Accuracy of driving
   - Did the car deviate from the drivable space?
   - How frequently were rules of the road broken?
3. Timely reaction to stop signs and traffic lights.
4. Avoidance of obstacles.

Ranking system is time based. Your team will obtain a time penalty based on the following constraints:

Time penalty	Description
+5 seconds	Every time car leaves the road
+10 seconds	Consistent departure from the road Every traffic violation committed

Table 1: Time penalty breakdown

Submission Requirements:

1. For **Stage 1** the code **must** be completed using the **Python 3.11.4** installer in the research resources .
   a. Should you make it to **Stage 2**, your code must be compatible with **Python 3.6.9**.
2. Maximum 1 minute video demonstration of code successfully completing a lap on virtual environment.
3. Submission must provide the following:
   a. Software: **GitHub** link to the repository with your team submission. Code will be run by a Quanser engineer to confirm successful implementation.
   b. Video: **YouTube link** demonstrating the result of your code.

Available Documentation:

1. Information for interacting with the virtual environment:
   https://qlabs.quanserdocs.com/en/latest/

2. Current examples released for the Self-Driving Car Studio:
   https://quanserinc.box.com/shared/static/84sb0kqgakrtlt6og39xrbgbaxp7x12z.zip

3. Competition webpage:
   https://www.quanser.com/community/student-competition/2024-student-self-driving-car-competition/

4. Additional software:
   2024 ACC Self-Driving Car Student Competition Resource Center

Getting Started

1. A registered team will be asked to create an account to obtain access to Quanser Interactive Labs. If you have not done so already, please go to https://portal.quanser.com/Accounts/Login?returnUrl=/ and create an account.
2. Use https://portal.quanser.com/Downloads to download and install QLabs on your system.
3. Use link #2 from the list of available documentation resources. Your system will download a .zip called research_resources. Follow the instructions provided in the README.txt to configure the required system environment variables. This setup sequence was designed to be run on **Windows 11.**

Configuring the Virtual Environment:

Step 1 – Use link #4 from the list of available documentation to download the latest files required to configure the virtual environment.

Step 2 – With a valid account please navigate to the open world. Once inside please select the warehouse workspace.

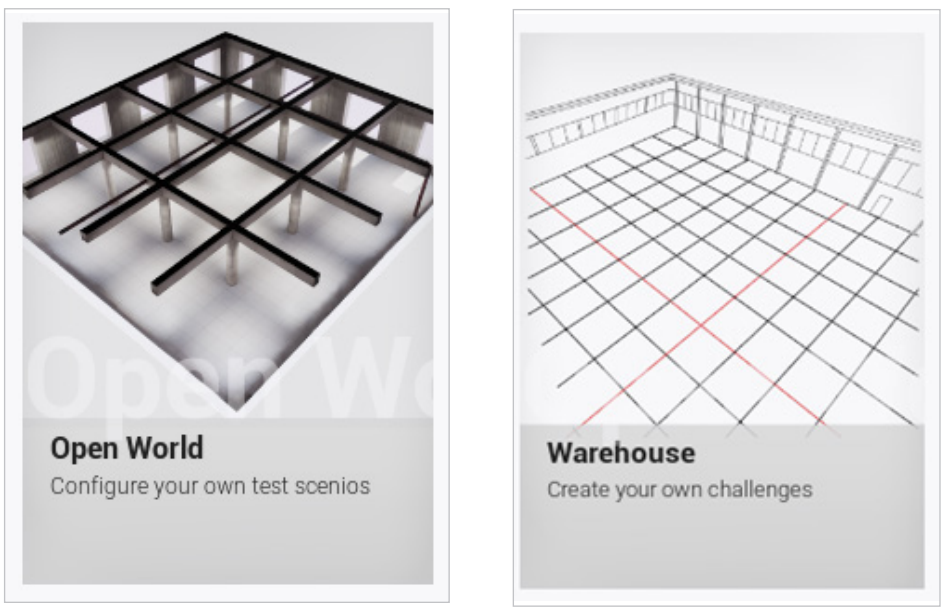

Figure 2: Quanser interactive labs menu item and warehouse workspace icons

Step 3 – In a terminal session run **Setup_Competition.py** to spawn the QCar in the virtual world.

Note: Please **do not** modify the Setup_Competition.py. Quanser will be using the original file to configure the workspace.

Step 4 – When ready run **Traffic_Lights_Competition.py** to enable control for the traffic lights. The fully configured environment should look as follows:

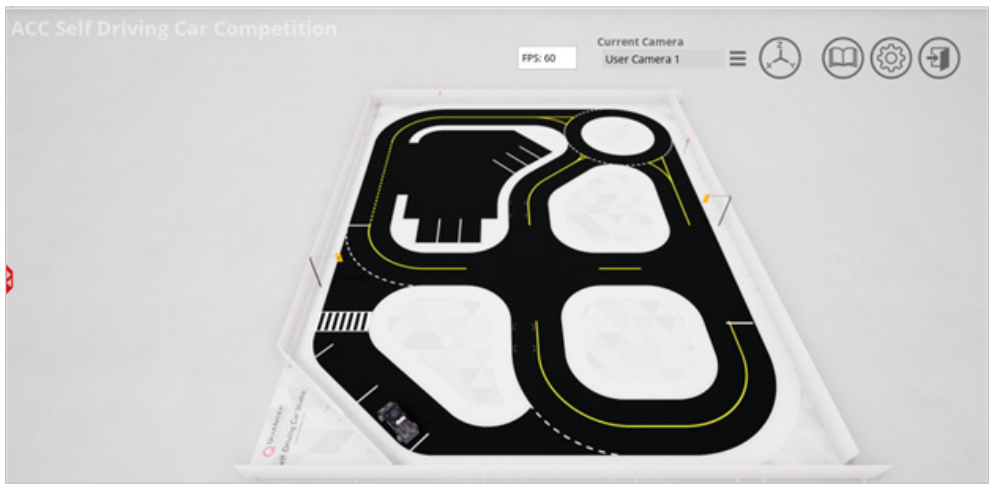

Figure 3: Competition virtual environment.

If your virtual environment looks like Figure 3 you can get started!

Suggested starting points:

1. Setting up the research studio resources on your system will create a Quanser folder int the following directory **C:/users/<User>/Documents/Quanser**
2. The folder labeled **SDCS/skills_activities/vehicle_control** contains an example of how to perform speed and steering control using the QCar.
3. Please use the **QCar** class definition found under **pal.producst.qcar** to interface with the QCar.

Good luck!

Competition Schedule

Jan 10 – Feb 29th	Stage 1 Open Registration
April 30	Stage 1 End Date
June 30	Stage 2 End Date
TBD	Stage 3 Guidance and Information