2025 ACC Self-Driving Competition

Stage 1 Competitor Guide

ACC SELF-DRIVING COMPETITION

In recent years, Self-Driving has become increasingly prevalent around the world. With the advent of Self-Driving cars hitting the roads, the demand for research behind Self-Driving is at an all-time high and algorithms need to be validated for safety and reliability. There are an infinite number of situations encountered daily by drivers that can vary drastically. An algorithm needs to be able to identify the scenario and accurately respond with a correct behavior. At their 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. The ACC competition presents a great opportunity to showcase world-class driving algorithms running in a safe and approachable manner at a 1/10th scale and in a virtual setting.

Competition Overview

In this year's competition, teams will try to maximize their profits as an autonomous taxi service. They will need to navigate through the city to selected pick-up and drop-off coordinates. During the trip, they will encounter different traffic scenarios that they will need to solve while adhering to the rules of the road. Depending on their performance during the trip, the teams will receive a rating for the executed ride that will influence the amount of money earned for thecompletion of the ride. Students will then receive additional ride requests and will try to maximize the amount of money they earn within a fixed timeframe.

The three stages of the competition are outlined below:

- **Stage 1: Virtual Design and Submission**
- **Stage 2: Implementation of Self-Driving algorithm on the Physical QCar 2**
- **Stage 3: On-site Competition**

All registrants will gain access to the Stage 1 virtual design phase, where a team's skills will be tested in Quanser's virtual environment. **The top 6 teams, based on selected criteria, will be chosen to move on to the second and third stages of the competition.**

In stage 2, the selected teams will receive a physical QCar 2 from Quanser and implement their Self-Driving algorithms on the physical QCar. They will then be **required to bring their QCar 2** to the competition venue and compete live against other selected teams.

More details will be released about stages 2 and 3 in a subsequent handbook.

Competition Schedule:

	Description
January 1st	Stage 1 Registration Opens
February 1st	Stage 1 Begins
April 30th	Stage 1 Ends
May 7th	Teams will be selected to move onto Stage 2
TBD (in July)	Stage 3

Stage 1: Virtual Design and Submission

Overview

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

Teams will receive access to the QCar 2 digital twin, available through Quanser Interactive Labs (QLabs). The digital twin provides access to all the same sensors and actuators that are on the physical QCar 2. These sensors will behave in the same manner that the physical does and are accessed through the same APIs that the physical uses.

Teams will be provided with basic scenarios to test their Self-Driving algorithms. These scenarios will serve as a starting point, but it is encouraged to develop more complex scenarios to truly highlight the capabilities of your Self-Driving algorithm for your video submission. Custom environments can be created in QLabs, so we encourage teams to get creative!

Teams will be ranked using the following criteria:

1. Readiness of a Self-Driving algorithm based on the core principles as stated in the Core Principles of Self-Driving section.
2. Accuracy of driving (staying within the lanes).
3. Timely reaction to road signage and traffic controls while adhering to traffic laws as stated in the Traffic Controls Rules section.
4. Clear and concise communication of Self-Driving concepts demonstrated in the video.

Core Principles of Self-Driving

Data Collection:

A Self-Driving algorithm must be able to collect and filter information from interoceptive and exteroceptive sensors. Demonstrating the conversion of raw data to meaningful information is critical for Self-Driving cars to make higher-level decisions during an autonomous task.

Interpretation:

Using system-relevant data, a Self-Driving car must correlate the gathered information to factors happening internally or externally in the environment. Examples of external factors include the identification of traffic signs, traffic lights, pedestrians, and other cars. Examples of internal data include battery monitoring and system state identification.

Control Systems:

From the set of viable options determined in the interpretation of the world, the car must be able to execute accurately on the chosen option. This includes staying within lanes, executing turns, stopping at traffic controls, altering a path based on an obstacle, and maintaining a desired speed.

Localization and Path Planning:

For a car to arrive at pick-up and drop-off locations, it must understand where it is within the roadmap. This may involve storing a global or local map of the environment in memory. A successful driving algorithm should be able to determine where it is in space and how to get to another location on the competition roadmap. It must also be able to adjust the selected route based on information obtained on its trip such as vehicles on the road, road obstructions, and pedestrians entering/leaving the roadway.

Software Compatibility

IMPORTANT: For the virtual QCar, it is expected that teams will use Ubuntu 24.04 and the provided docker containers. Using other software is allowed but will not be supported.

- **Linux:**

  Teams are expected to have a computer with **Ubuntu 24.04** and an **Nvidia GPU**.

- **ROS:**

  The supplied ROS nodes use ROS 2 Humble, which is already configured through the Docker container.

- **Python:**

  The Python version installed within the Docker container is **Python 3.8.**

NOTE: We are leveraging Isaac-ROS Docker containers for the entire competition. It is expected that the Isaac-ROS Docker container provided will be used on the physical QCar 2 during Stages 2 & 3.

Submission Requirements

1. Follow the above Software Compatibility section.
2. If you are ONLY using Python you must interface with the QCar 2 via the QCar class in the pal library. Controlling the QCar 2 or gathering data via the qvl library functions will invalidate any submission.
3. Maximum 3-minute video demonstration of the Self-Driving capabilities and explanations.
4. Submission must provide the following:
   a. Software: **GitHub link** to the repository with your team's submission. The code will be run by a Quanser engineer to validate performance.
   b. Video: **YouTube link** demonstrating your code.

Technical Information

Getting Started

Follow Software_Setup section on the GitHub repository to fully set up your Linux system.

Available Documentation

1. Information for interacting with the virtual environment: QLabs Documentation
2. Python API Documentation: Python API
3. C API Function Documentation: C API
4. Current examples released for the Self-Driving Car Studio : QCar Resources
   a. Password is: acc2025denver
5. Competition webpage: Competition
6. Github

Stage 1 Virtual Scenarios

There will be 7 base scenarios that are provided for the virtual world as follows:

1. Setup_Base_Scenario_stop_ACC_Competition_2025
2. Setup_Base_Scenario_yield_ACC_Competition_2025

3. Setup_Base_Scenario_trafficLight_ACC_Competition_2025
4. Setup_Base_Scenario_trafficCone_ACC_Competition_2025
5. Setup_Base_Scenario_roundabout_ACC_Competition_2025
6. Setup_Base_Scenario_pedestrian_ACC_Competition_2025
7. Setup_Competition_Map

These scenarios serve as a basis for how teams can test their code. They can also be used as examples of how to spawn in different objects and create custom scenarios!

Tip: When designing custom scenarios, it is helpful to use the coordinate tool to spawn actors:

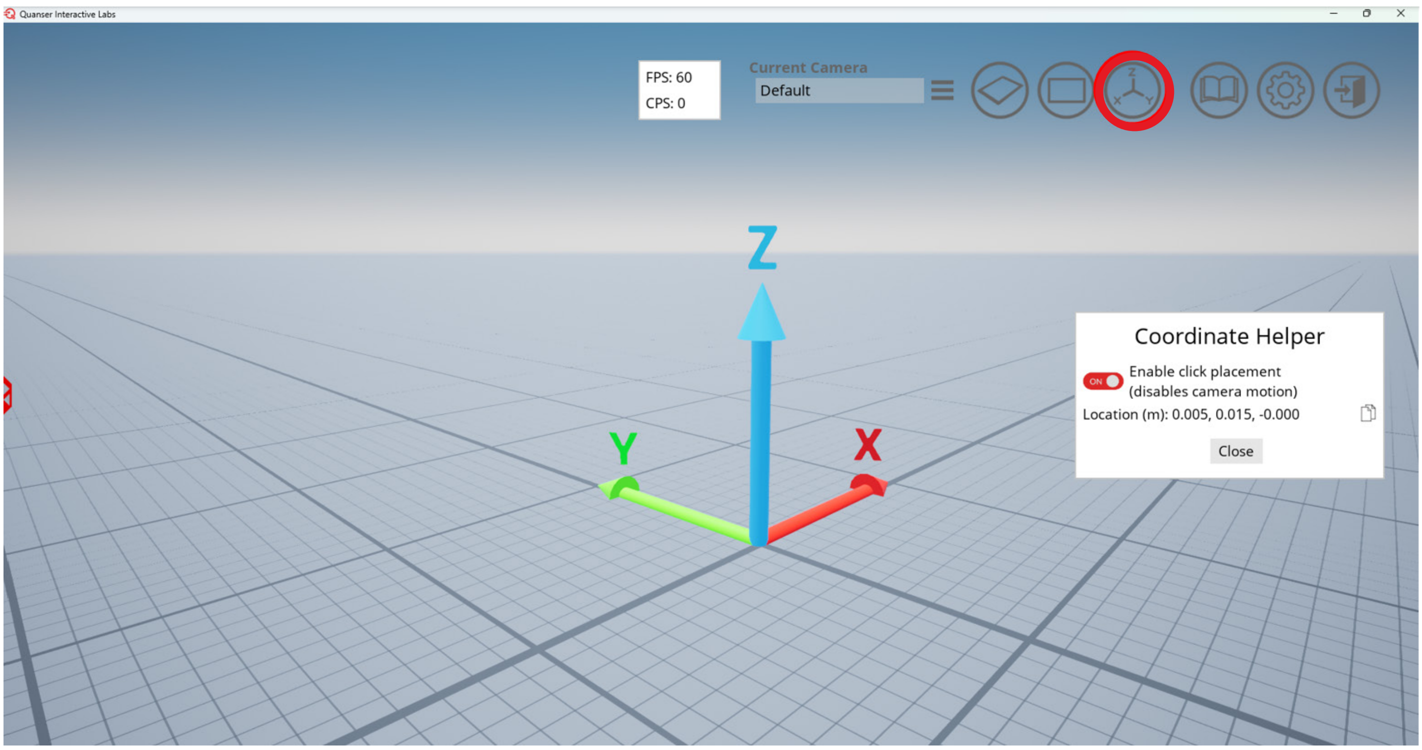

Figure 1: Coordinate Tool

Coordinate System

In the Stage 2 & 3 of the competition, a series of coordinates will be provided to teams indicating pick-up and drop-off locations. The coordinate system is defined as follows:

[0, 0, 0] and the orientation of the coordinate tool defines the base reference frame for all coordinates specified later in the competition.

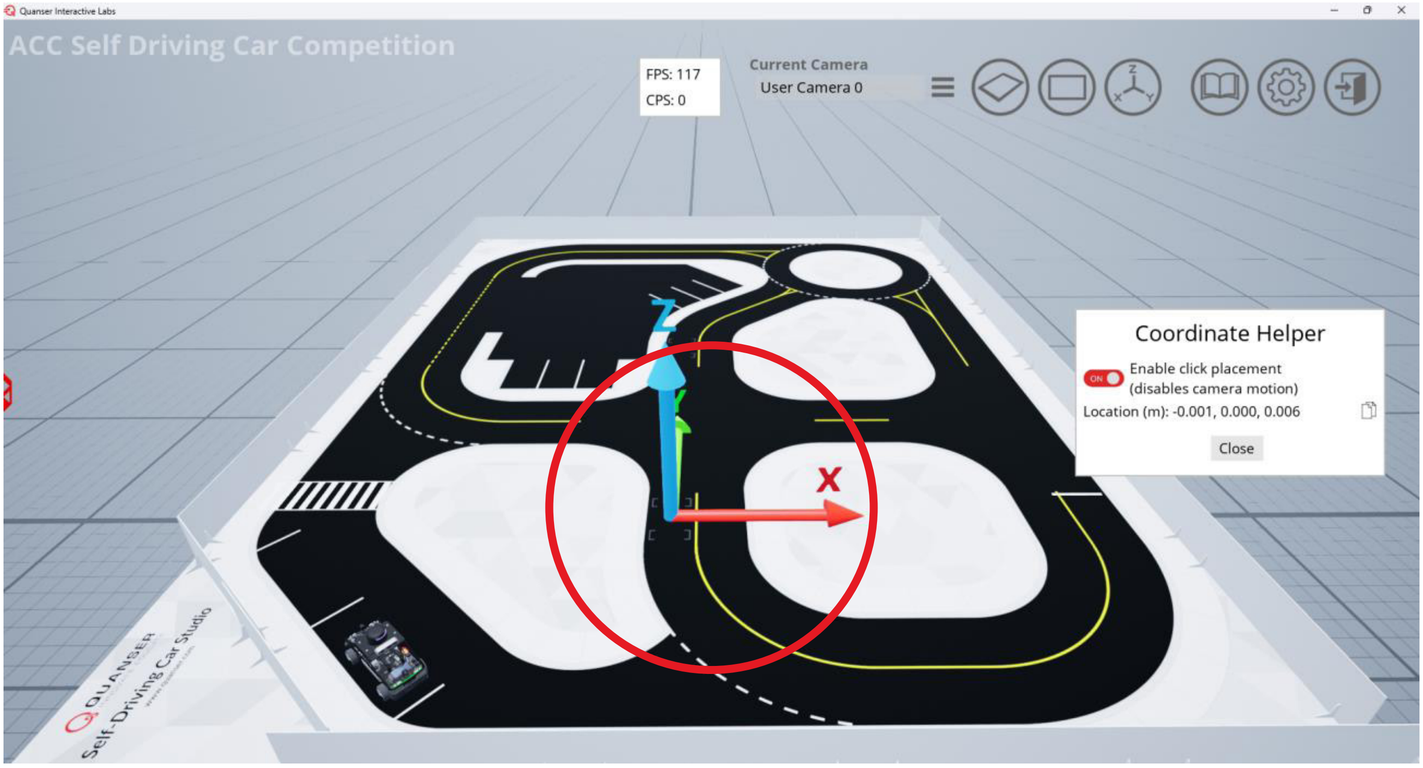

Figure 2: Coordinate system of the competition road map.

Figure 3: Origin of the Coordinate System

The competition roadmap in Figure 2 matches the scale of the roadmap in the real world. This means that the size of the QCar 2, the size of the roadmap and the placement of the origin will be identical to the setup that will be used in Stages 2 & 3 of the competition.

Traffic Controls Rules

All traffic control rules will be based on The Official Ministry of Transportation (MTO) Driver's Handbook which can be found here:

[The Official Ministry of Transportation (MTO) Driver's Handbook](#)

QUANSER
INNOVATE · EDUCATE