

# 2025 ACC SELF-DRIVING CAR STUDENT 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 or even slightly from one another. An algorithm needs to be able to identify the scenario and accurately conclude a correct behavior. 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. 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 the completion 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 selected teams, based on specific 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 1 <sup>st</sup>	Stage 1 Registration Opens
February 1 <sup>st</sup>	Stage 1 Begins
April 30 <sup>th</sup>	Stage 1 Ends
May 7 <sup>th</sup>	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 Requirements

**Important:** It is expected that teams will use Linux and ROS 2. Any other languages used are allowed but will not be supported.

### ➤ **Linux:**

If using Linux, teams are required to use **Ubuntu 20.04** since this is the Ubuntu version that is on the QCar 2.

### ➤ **Python:**

During the virtual stage 1 students must use **Python 3.11** since this is the Python version that is installed with the QCar 2 resources provided.

IMPORTANT NOTE: During stages 2 and 3, the Physical QCar 2 has **Python 3.8.10** installed on it. This version of Python **CANNOT** change. Ensure that your code in the Stage 1 submission is compatible with Python 3.8.10. Avoid packages that will not be compatible since they will not work using a physical QCar 2 in stage 2 and 3!

### ➤ **ROS:**

The ROS nodes supplied are for ROS 2 Humble.

## **Submission Requirements**

1. For **Stage 1** teams may use any of the following software:

- a. Python 3.11
- b. ROS 2 Humble
- c. Matlab/Simulink
- d. C++

The above software must comply with the Software Compatibility Requirements section above.

2. Maximum 3-minute video demonstration of the Self-Driving capabilities and explanations.

3. 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 the below steps to get set up for the competition:

1. Create a Quanser Interactive Labs account here: [Create Quanser Interactive Labs Account](#).
2. Use [QLabs Installation](#) to download and install QLABs on your system.
  - a. If using MATLAB use the following link to install QLABs: [Matlab QLABs Installation Instructions](#).
3. Download the SDCS Resources containing QCar 2 resources from here: [QCar Resources](#)
4. Explore!

### **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](#)

## 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

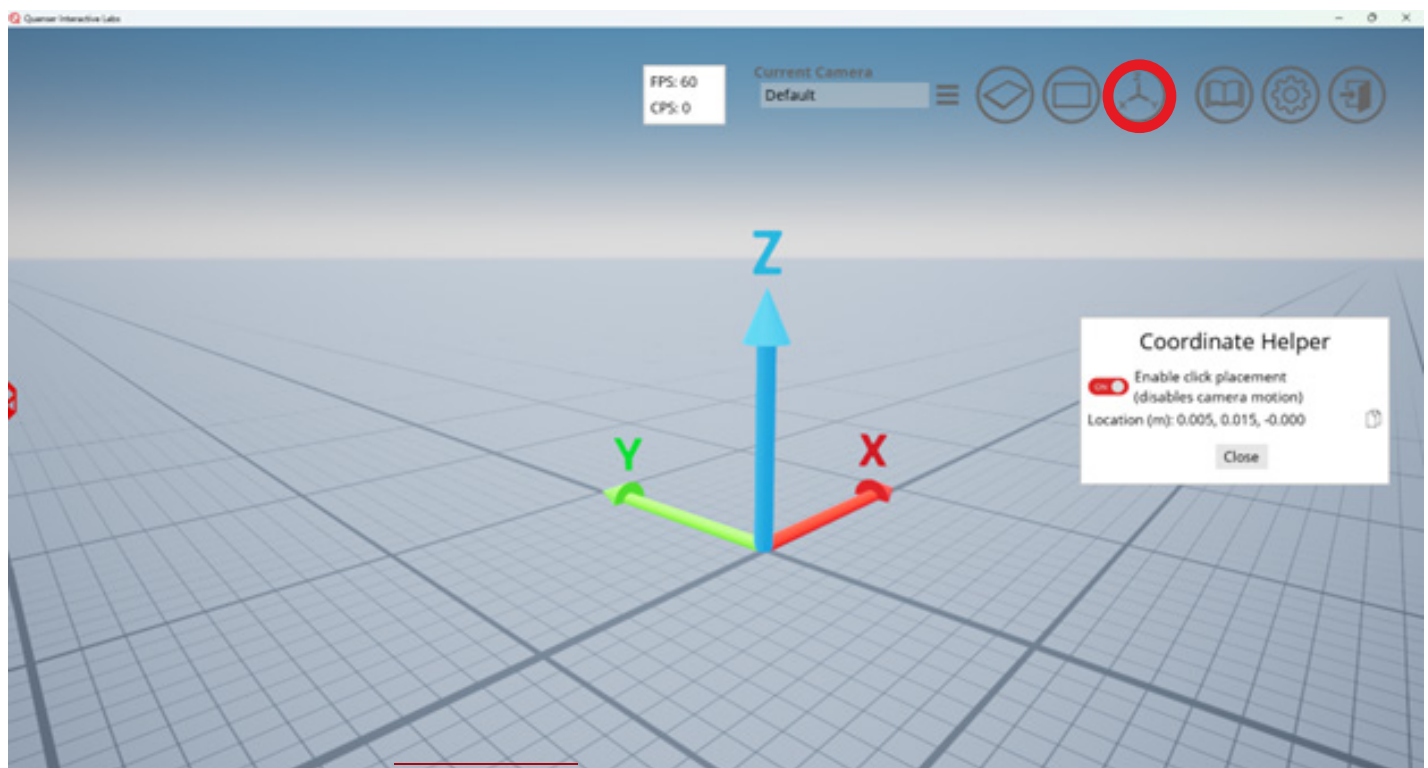


Figure 1: Coordinate Tool

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:**

## 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:

- The coordinate system's origin will coincide with the origin in the Quanser Interactive Labs Cityscape world:locations. The coordinate system is defined as follows:

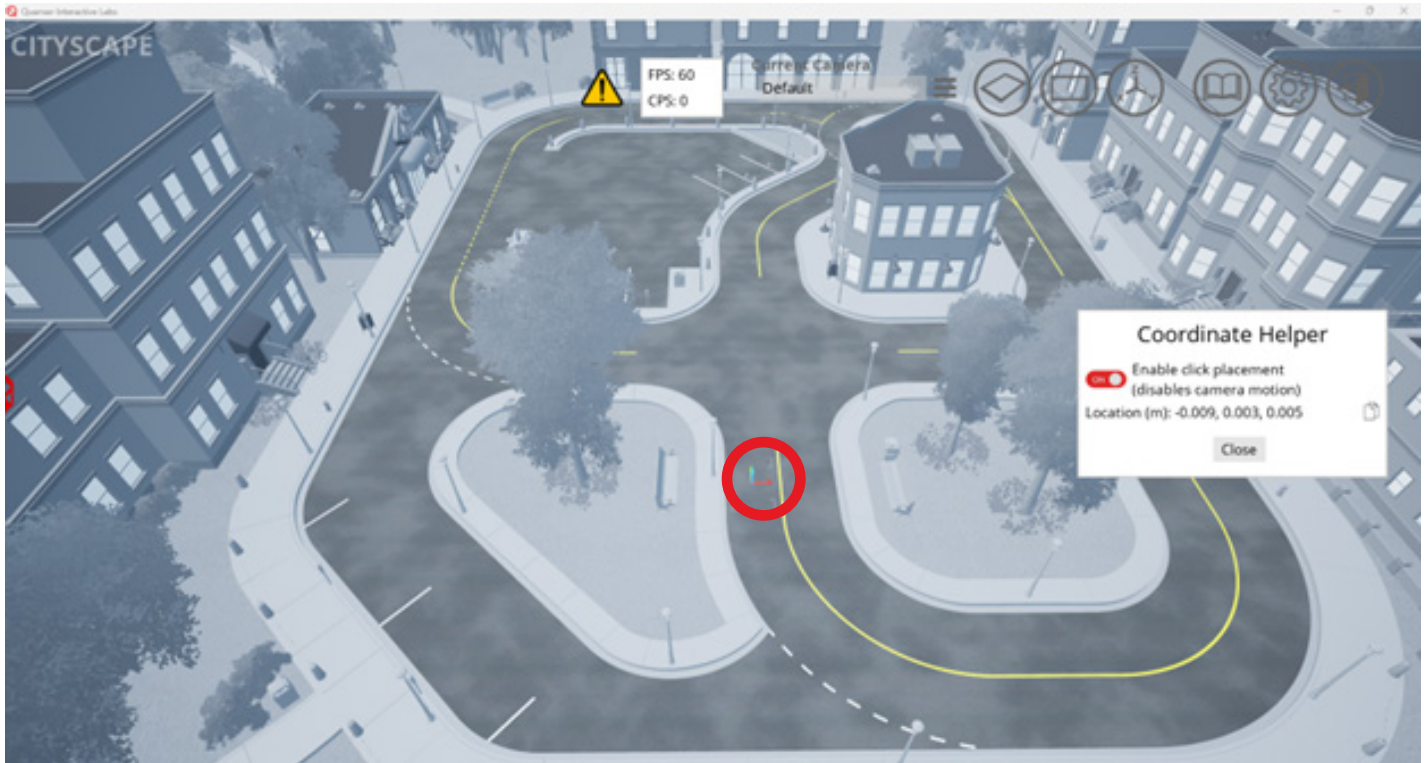


Figure 2: Origin of the Coordinate System

- The map used in this world is full scale, and a 1/10th scale map will be used for the physical portion 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](#)

