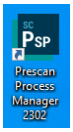
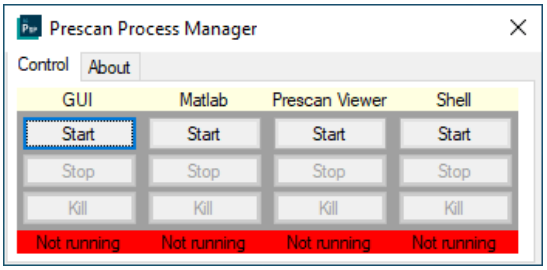
Step by Step Experiment Creation in Prescan

Assuming a successful installation and licensing of PreScan (version 2302) and the appropriate configuration of the associated MATLAB (version R2022b), you are set to start running Prescan seamlessly.

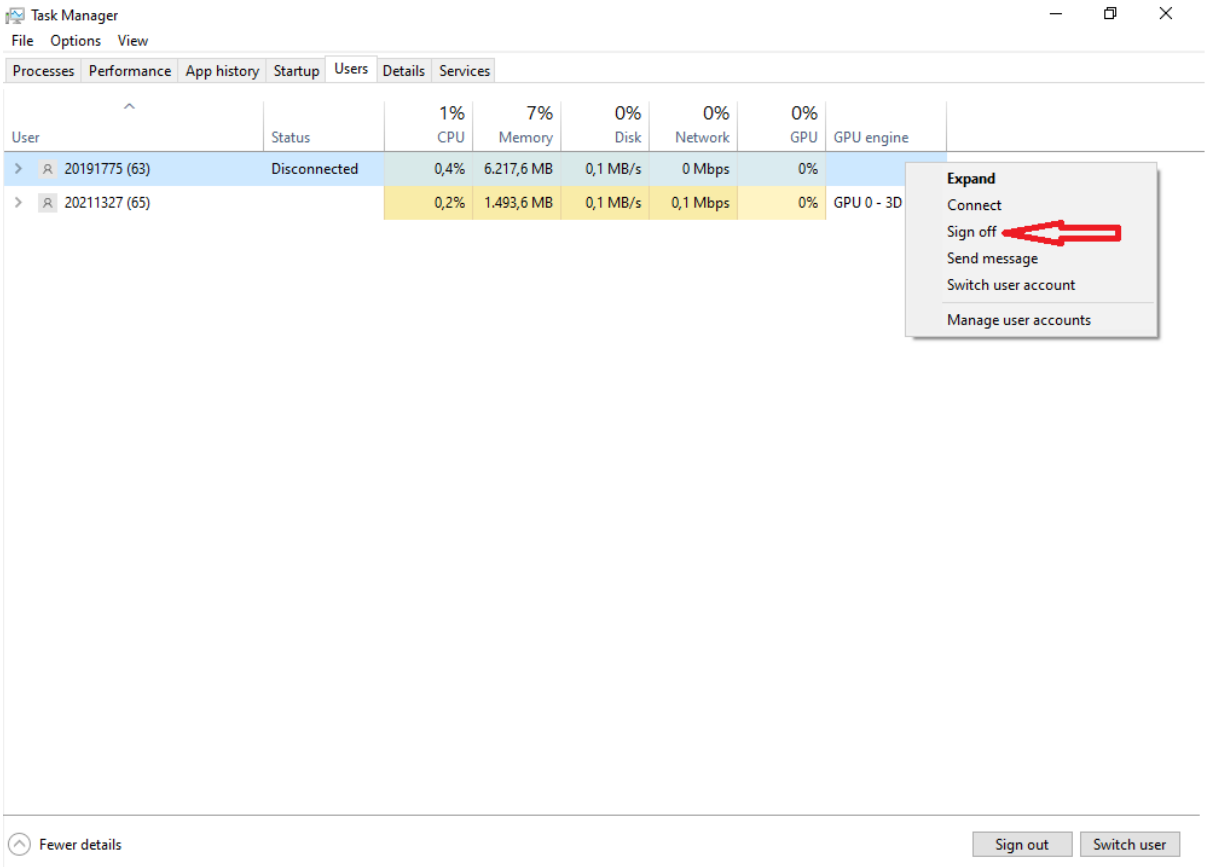
To launch Prescan, start the *Prescan Process Manager 2302*, 

which opens a window from where you can run several applications, including the Prescan GUI, MATLAB, Prescan Viewer, and a Terminal set at the Prescan directory.



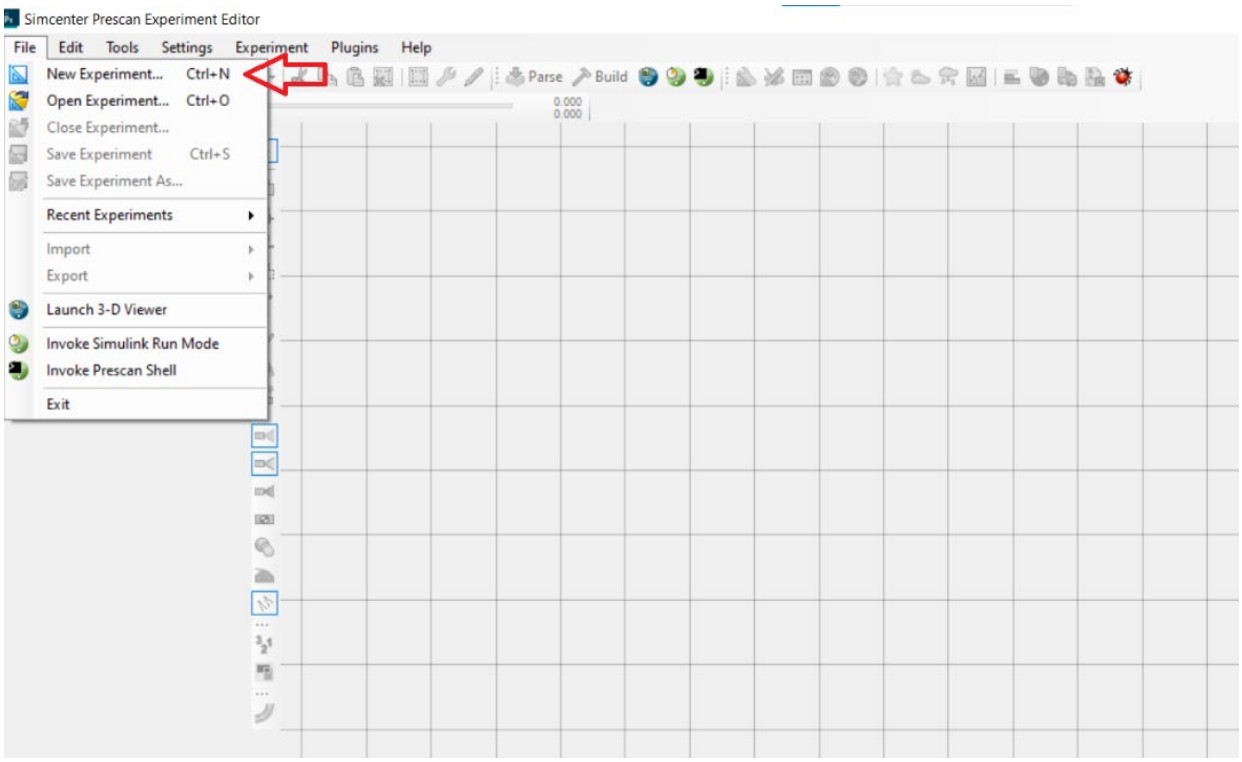
Remember, MATLAB needs administrative rights to access Pressman's initialization files. If you lack these rights, **ensure to run the Prescan Process Manager 2302 as an administrator.**

If you get an error message saying that Prescan is in use by another user you can simply sign them off and then try again

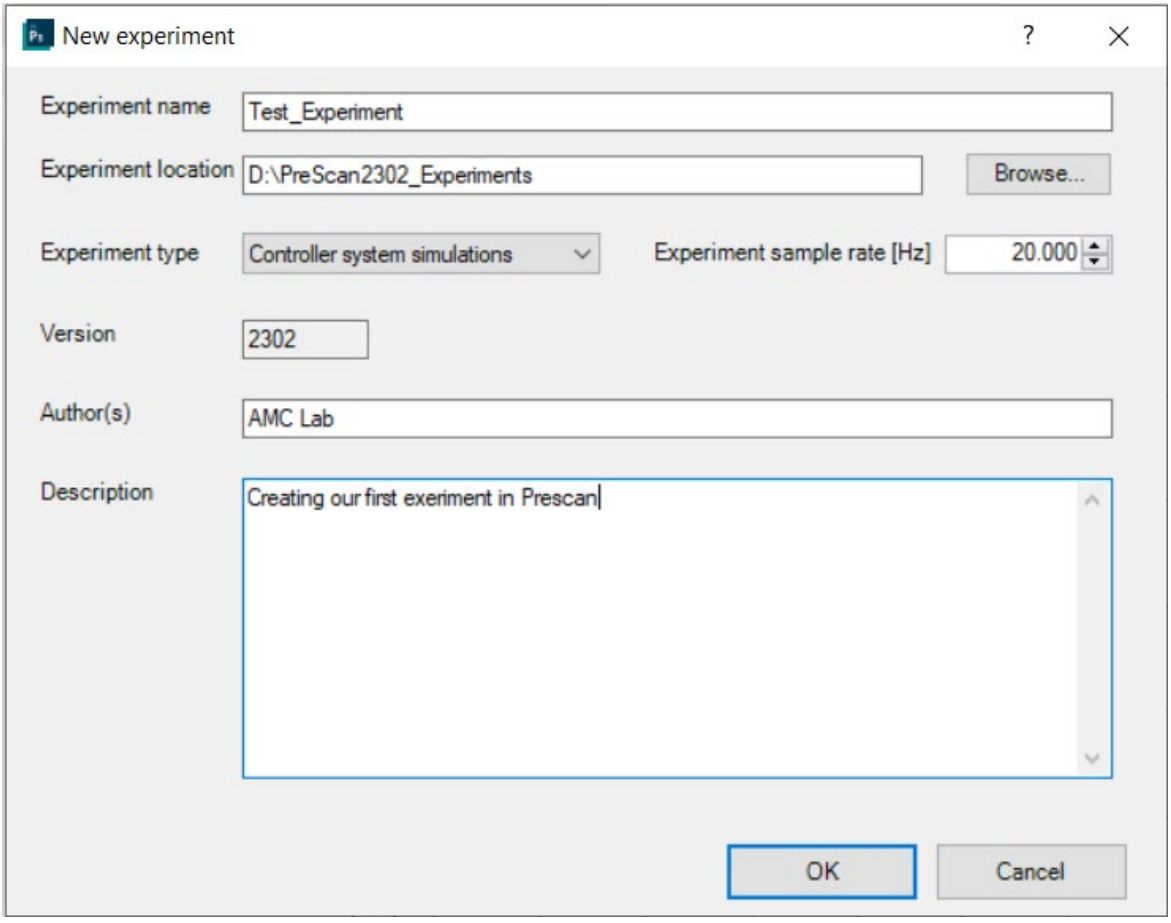


For this guide, since it only focuses on scenario creation in Prescan, we will only be launching the GUI and MATLAB.

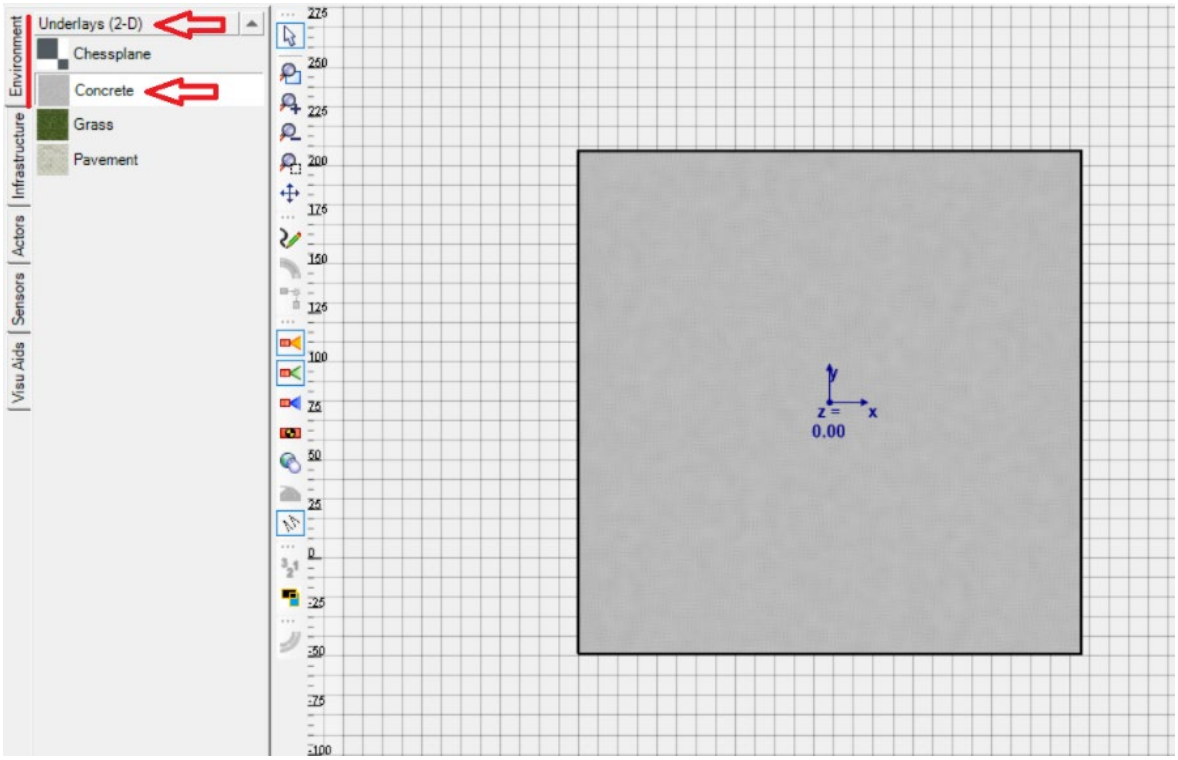
We start by creating new experiment and for that we go to *File* -> *New Experiment* :



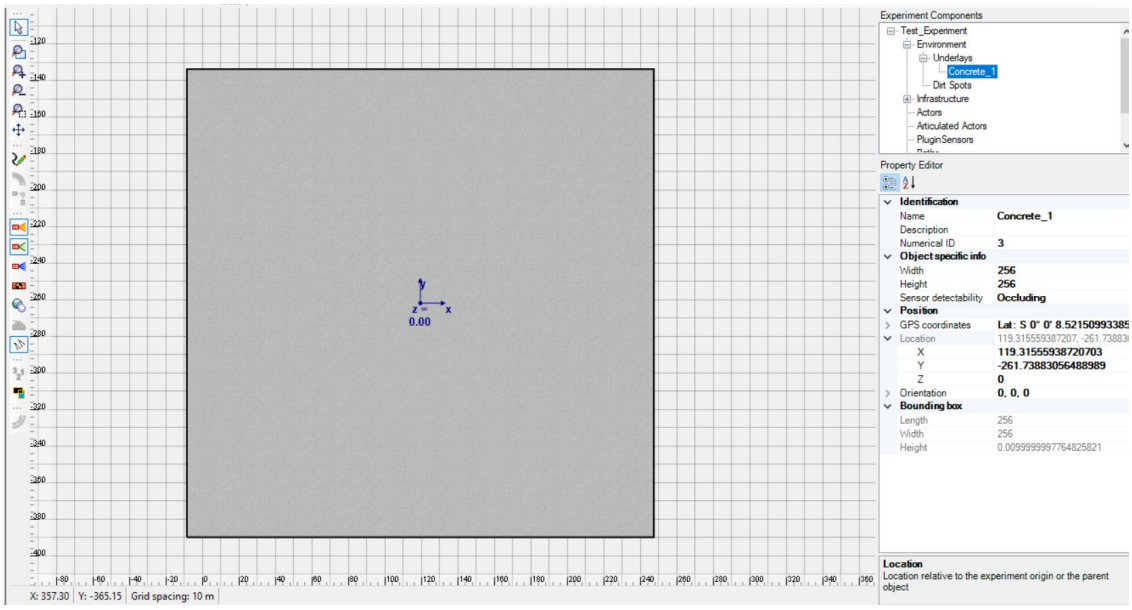
The we fill in the experiment information; name, author, and description:



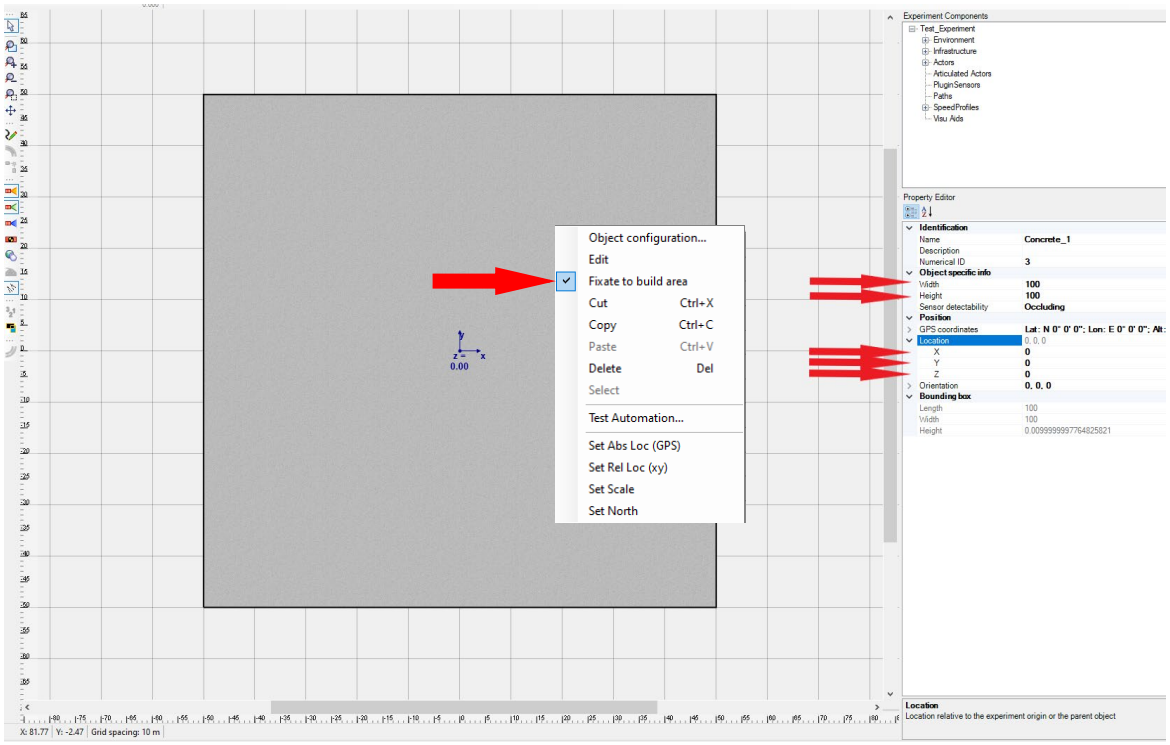
Now from the left-hand side library elements we choose *Environment* and then *Underlays* from there we drag and drop *Concrete* element.



Using the *Property Editor* on the right-hand side we can change the specifications of the concrete elements i.e. its dimensions, locations, orientation, etc.

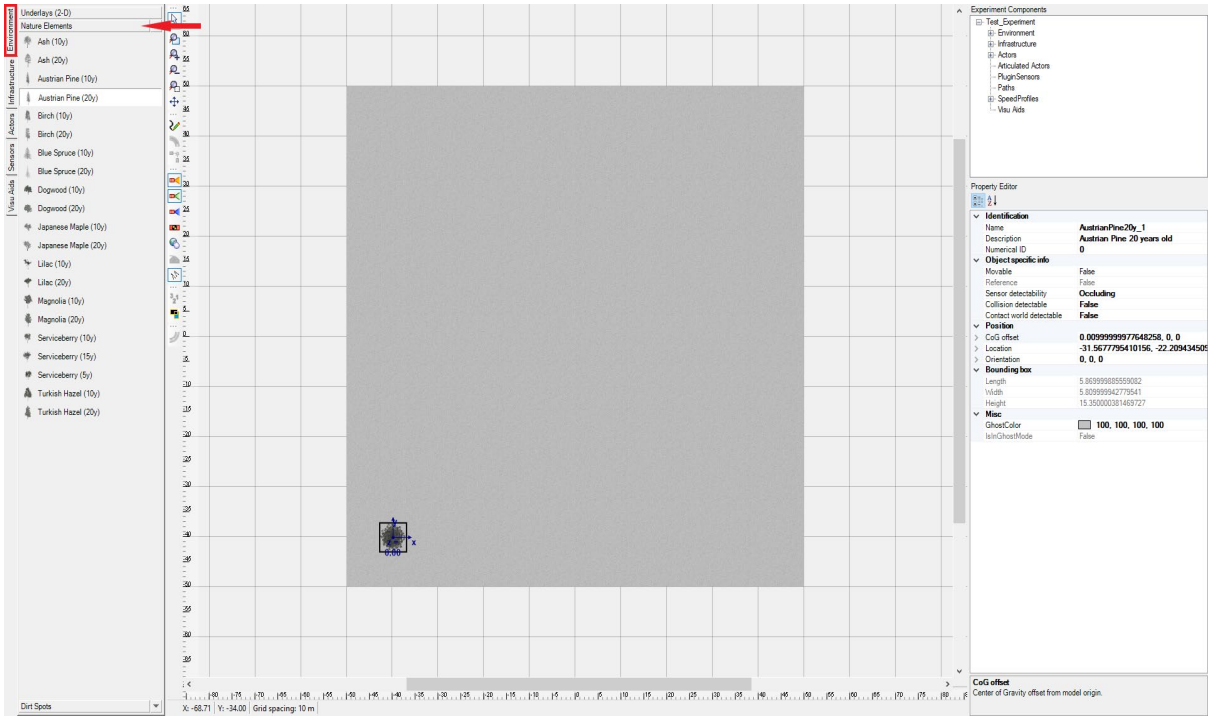


For the purpose of our experiment, we will set the *Location* coordinates to X=0, Y=0, Z=0 and the dimensions to *Width* = 300, *Height* =300.

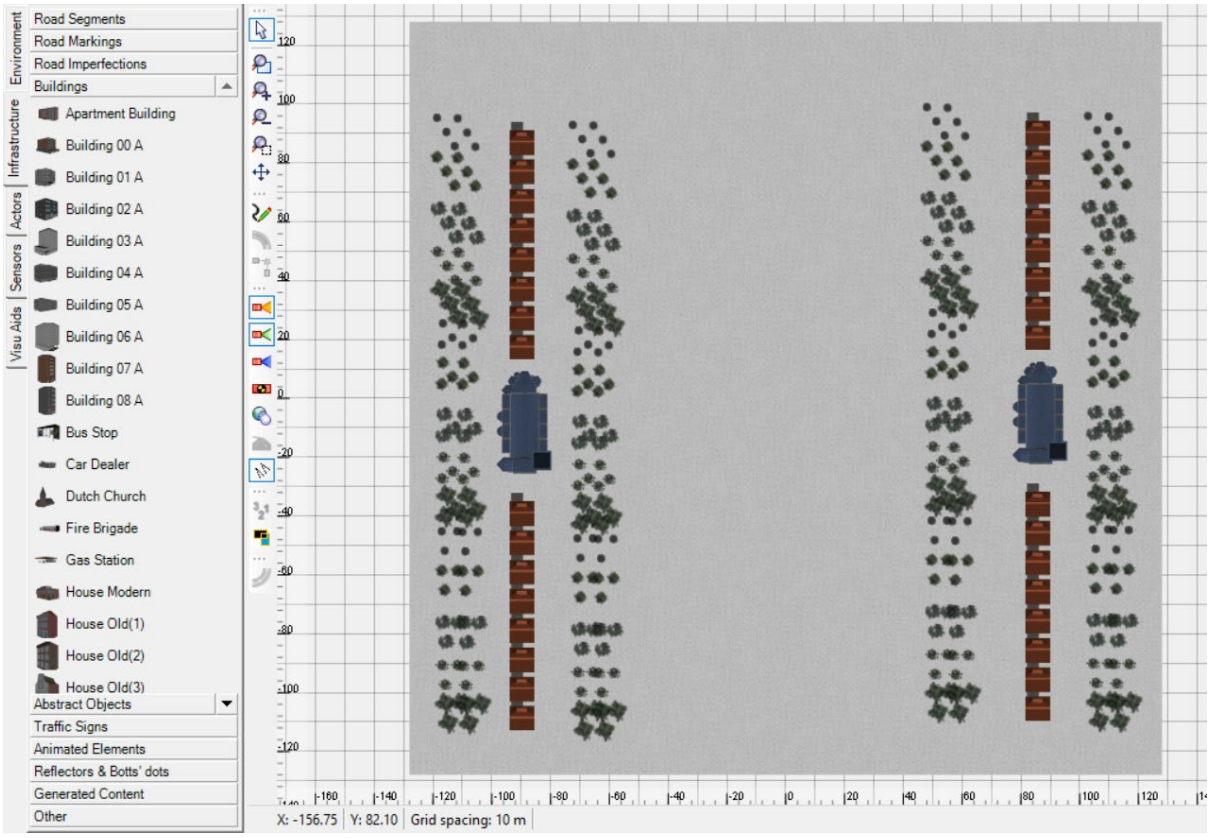


Now we will add some trees and for that we go to the *Environment* tab we go to *Nature Elements* (*Austrian Pine (20y)* for example):

Tip: To avoid mistakenly moving the object in the design process we can fix the object to the build area by right clicking on the object and choosing *Fixate to build area* option

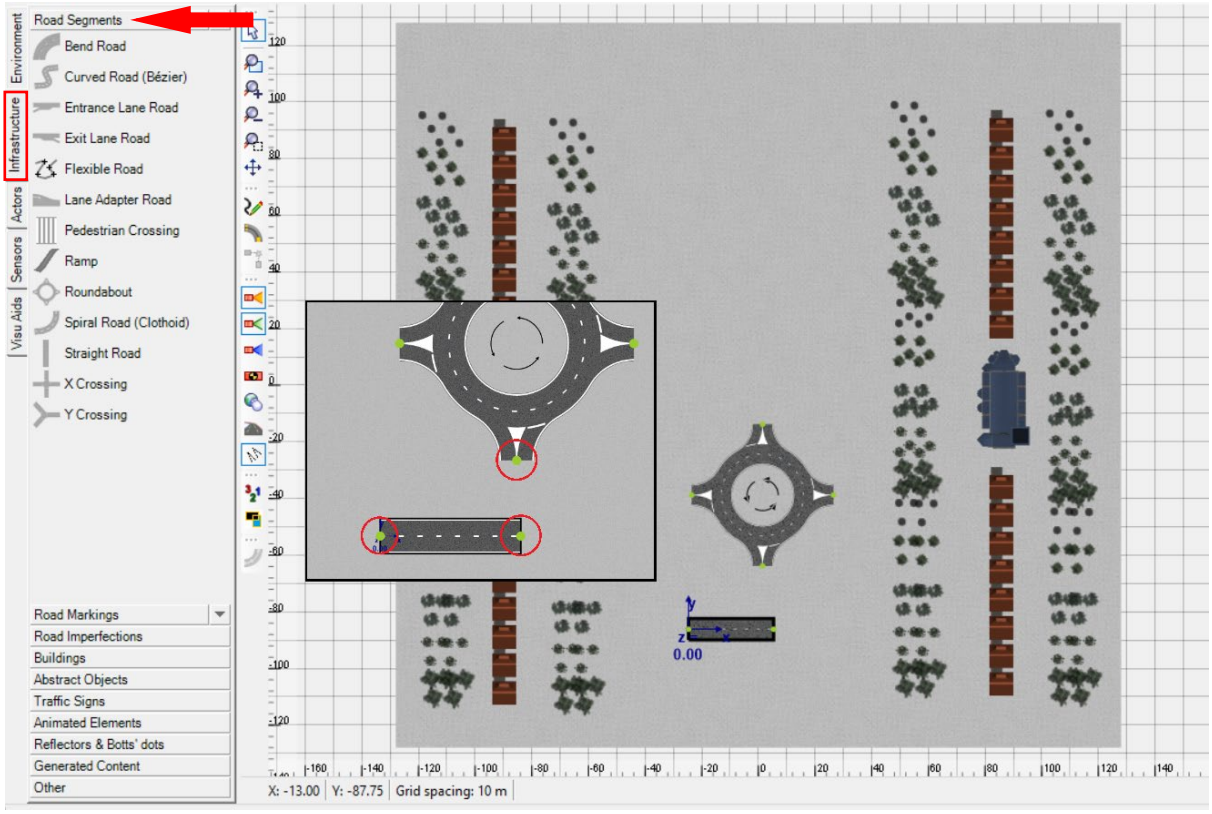


Now from the *Library Elements* we go to *Infrastructure,* and we add some *Buildings*



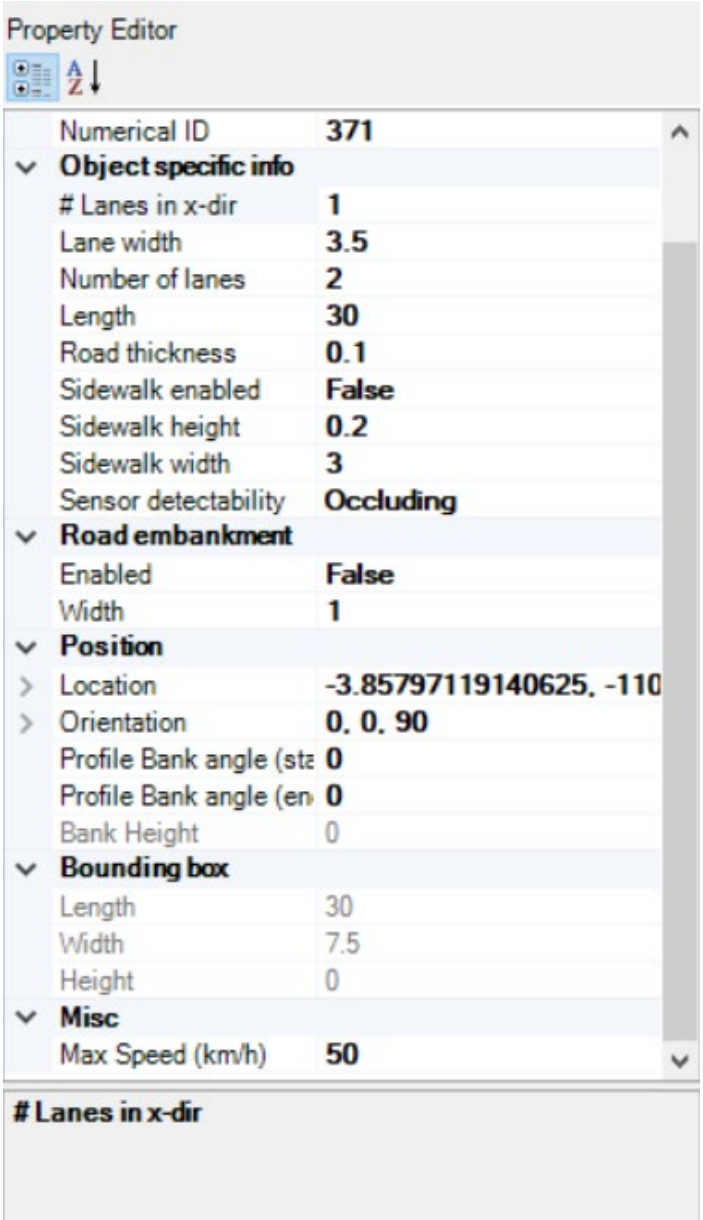
Tip: We can select several objects at the same time and then copy and paste them for easier and faster design.

Now we start creating the road for our actor from the *Road Segments* list in the *Infrastructure* tab:

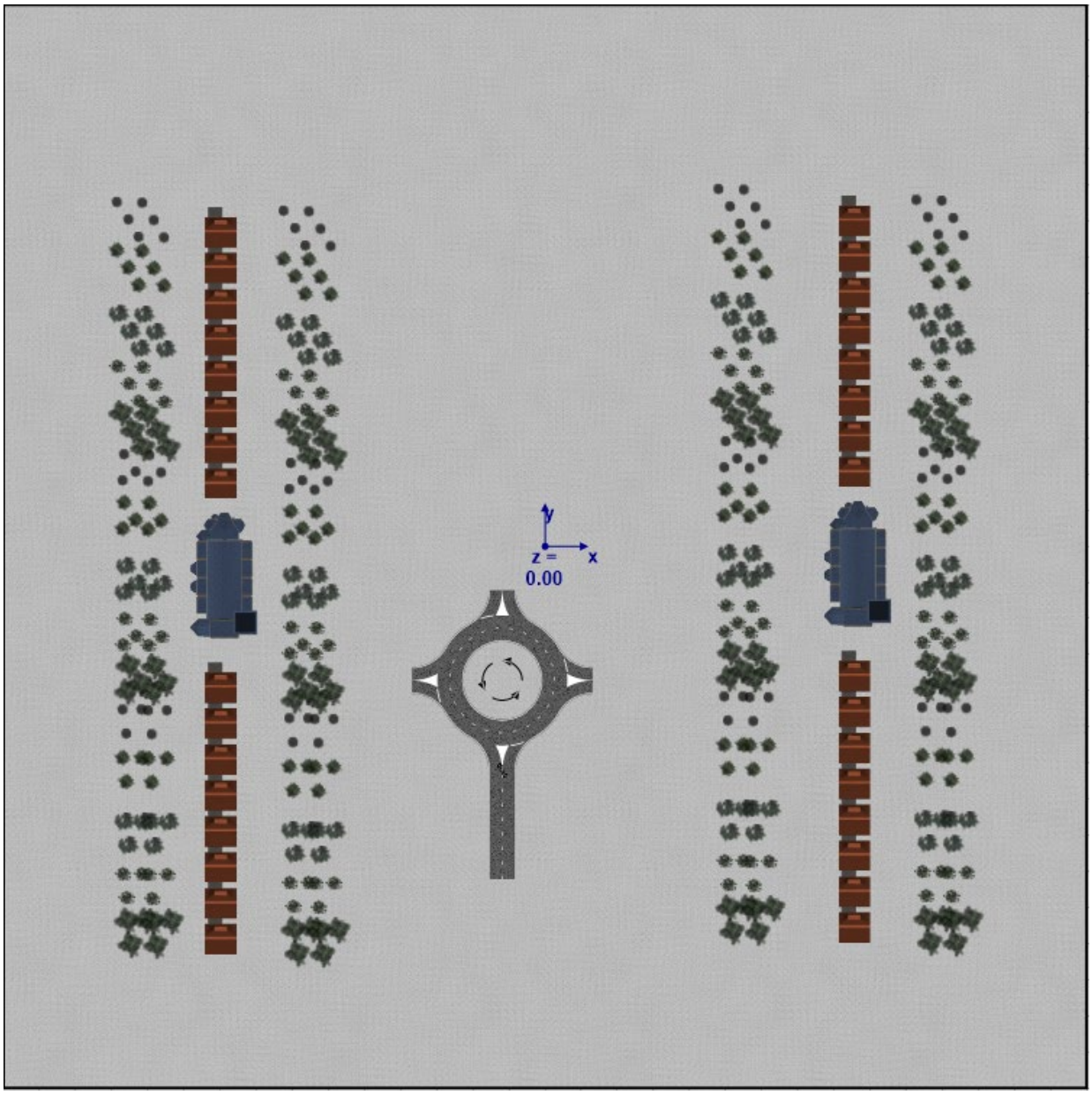


We added a straight road and a roundabout, we can change the orientation of the road segment from *Property Editor* on the right.

For each road segment we can specify the number of lanes, lane width, adding a sidewalk, maximum speed.



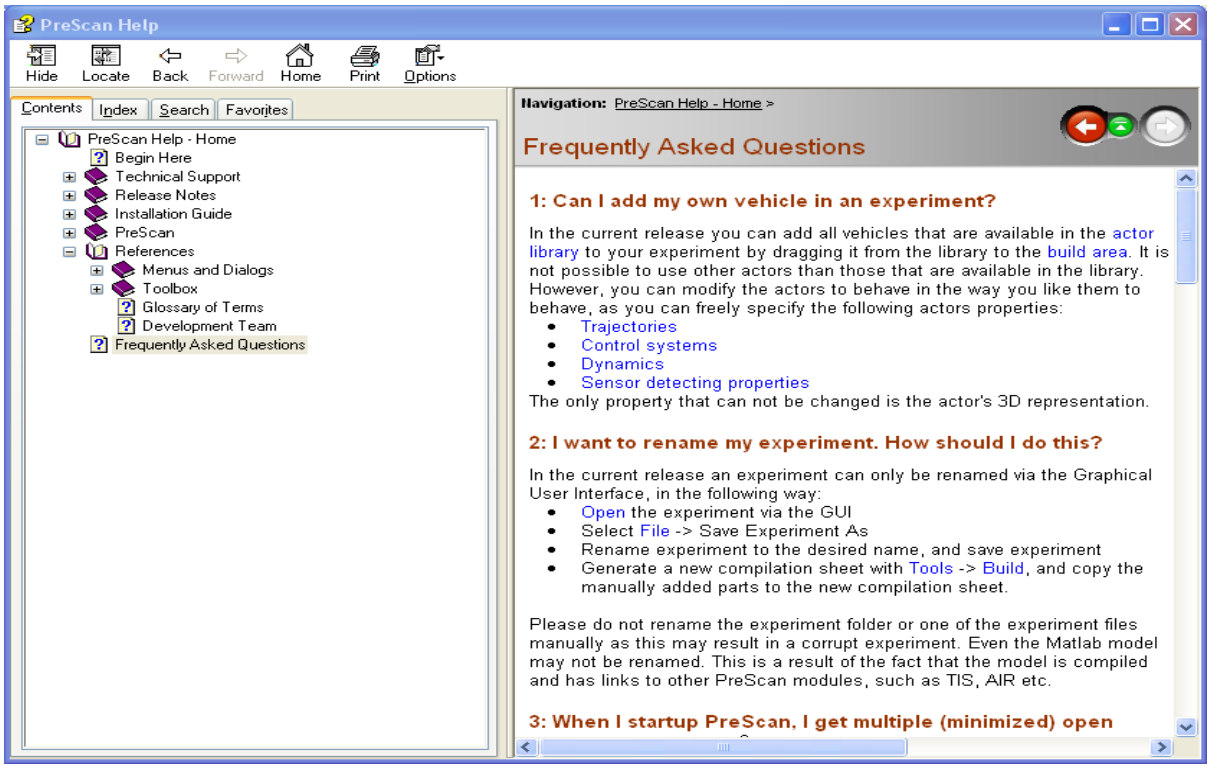
However, when moving the road segment by holding the right click and while dragging the object a small lime-green dots will appear when connecting these dots the road segments will be automatically aligned according to the orientation of the other segment.



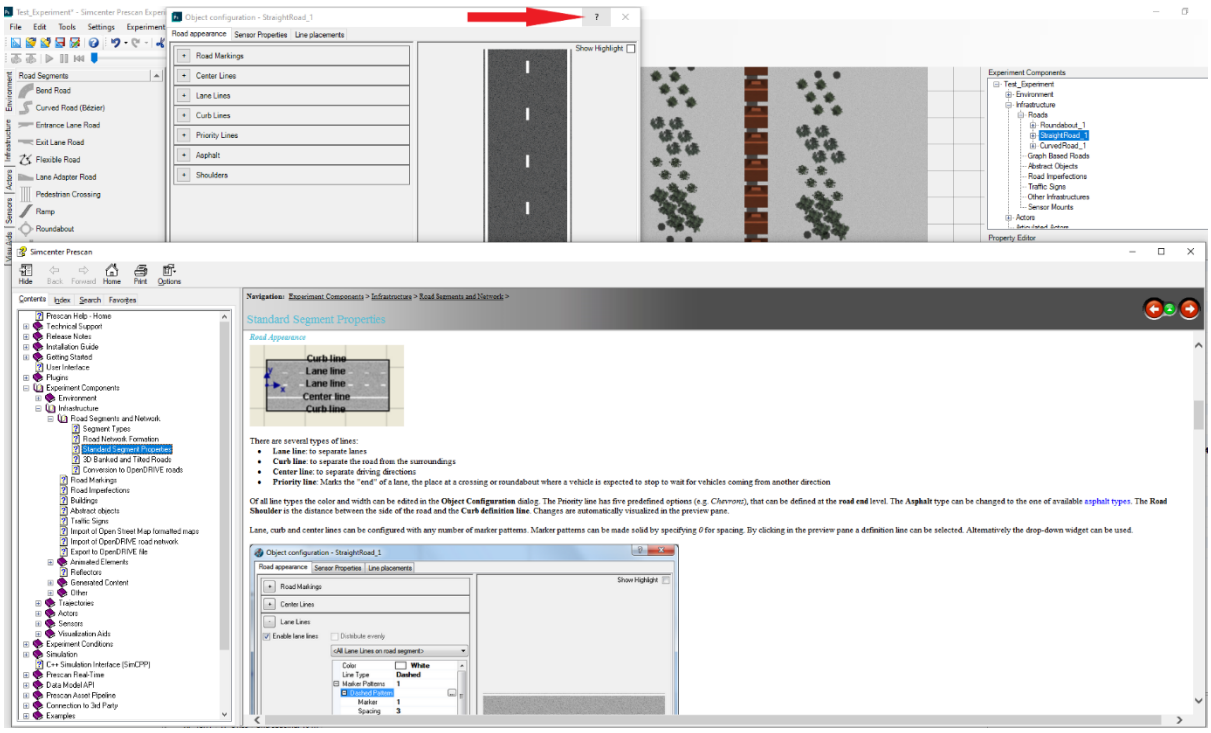
Using different combinations of road segments, we can create some complicated road.

Note: For more advanced design of the road we can use the *Help Center* in Prescan:

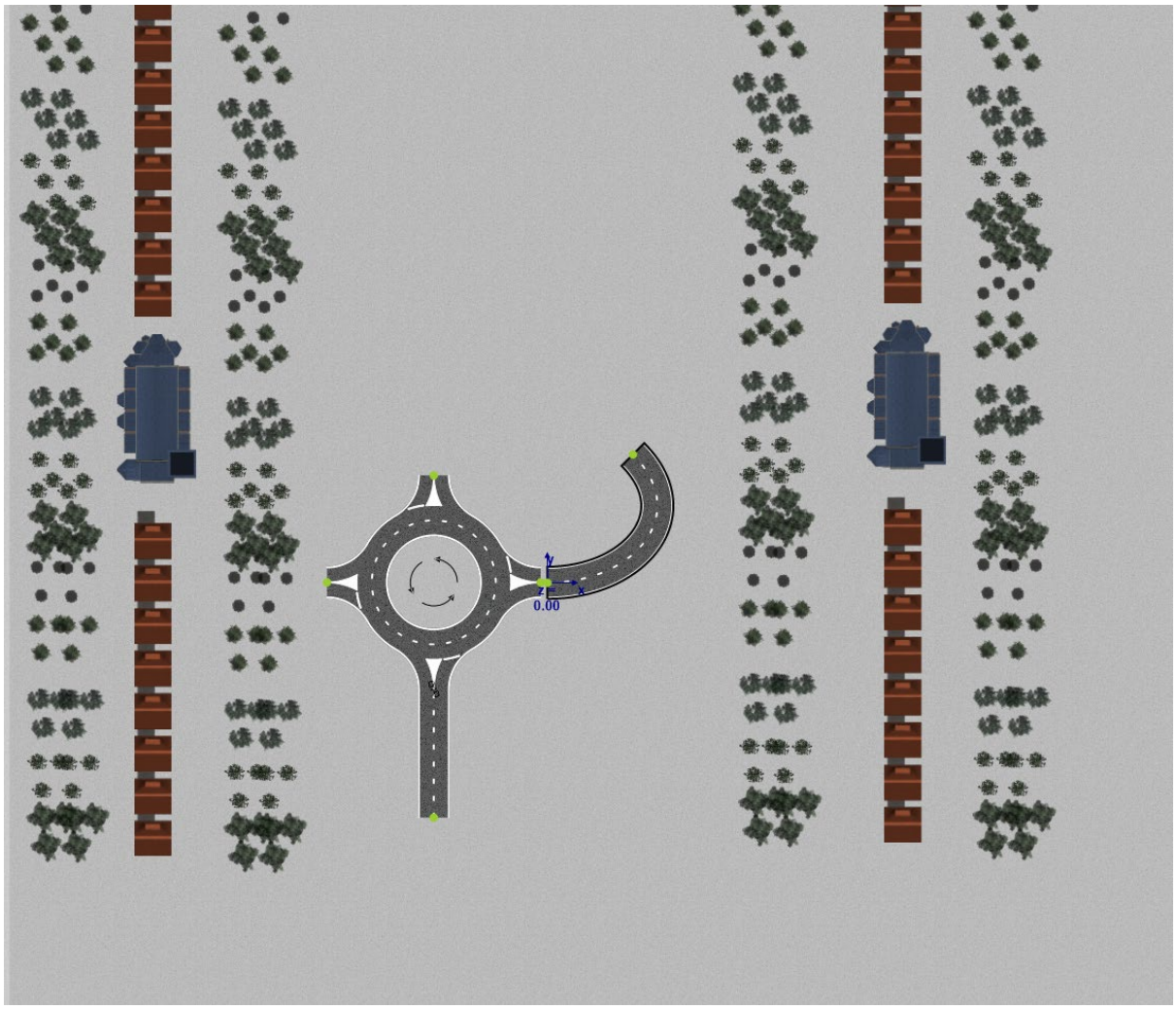
PreScan provides comprehensive assistance through its help file. It comes with a Frequently Asked Questions (FAQ) list and a navigational tool that allows you to browse like you would on an internet browser, with options like 'back', 'forward', and 'home'. It is equipped with numerous search terms to simplify finding information on relevant subjects.



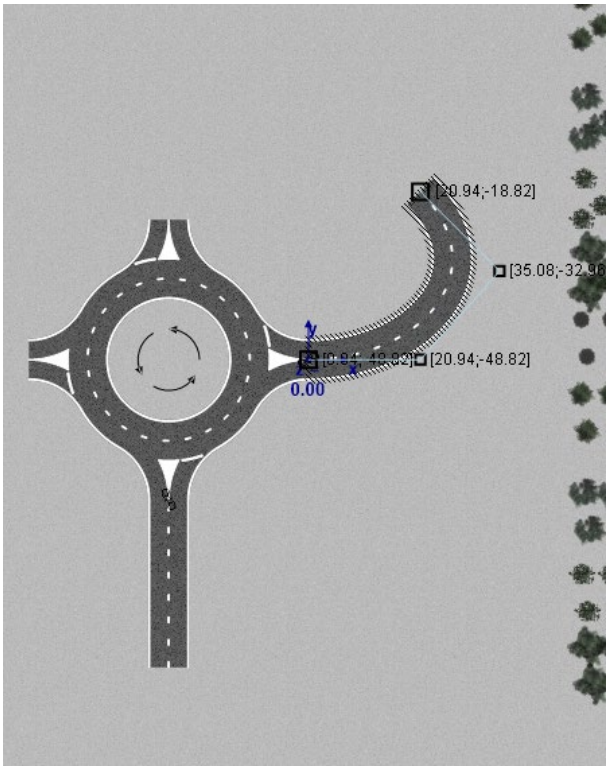
Help can be accessed by clicking on the "?" on any dialog. Upon doing so, PreScan's help will automatically display documentation related to the active topic. This ensures immediate and relevant support as you work through your tasks.



No we will close the road, we can do that by adding for example a *Curved Road (Bézier)*



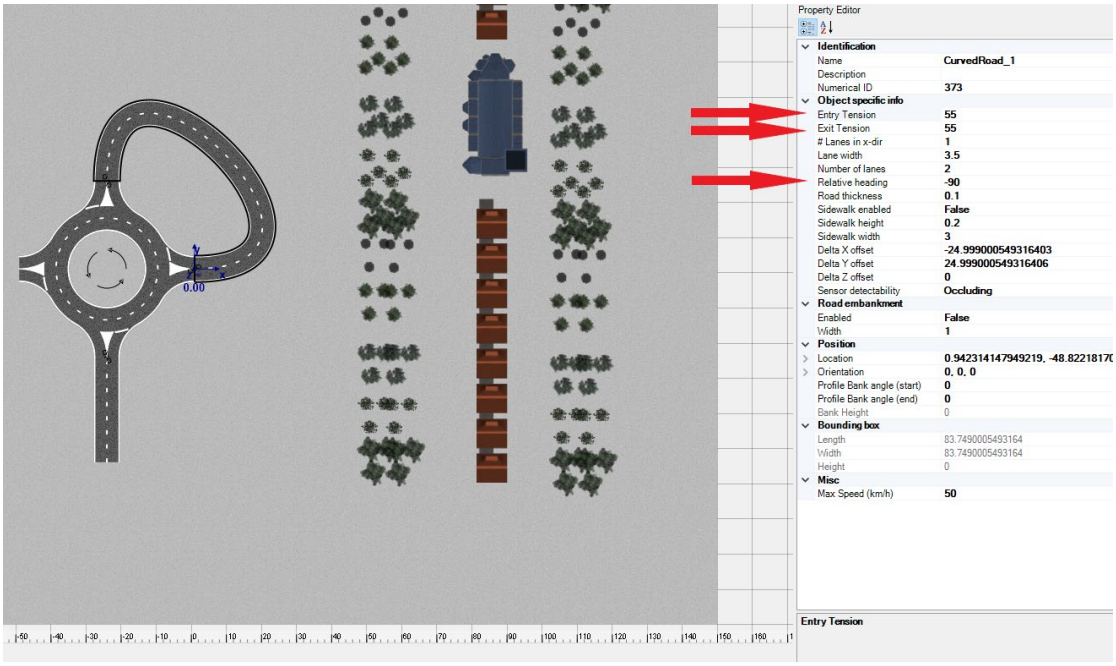
now we try to close the road by right clicking on the *Curved Road* segment -> *edit* and the we can modify the length and curvature points:



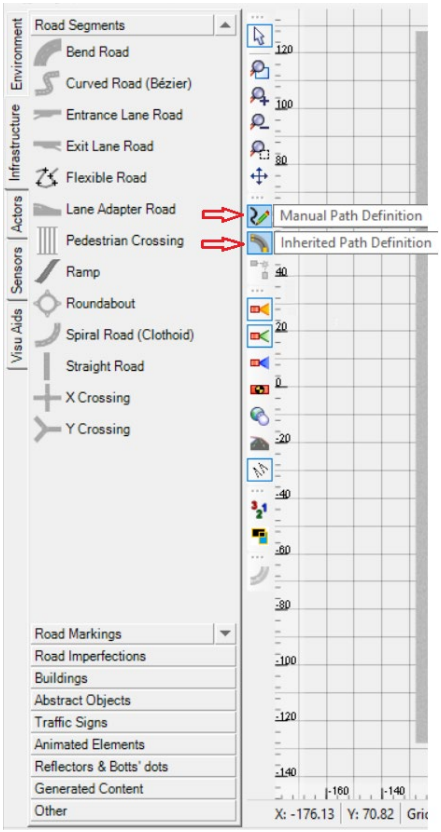
Or we can modify the sepcifications in the *Property Editor* as follow:

Entry Tension = 55, Exit Tension = 55, and Relative Heading = -90.

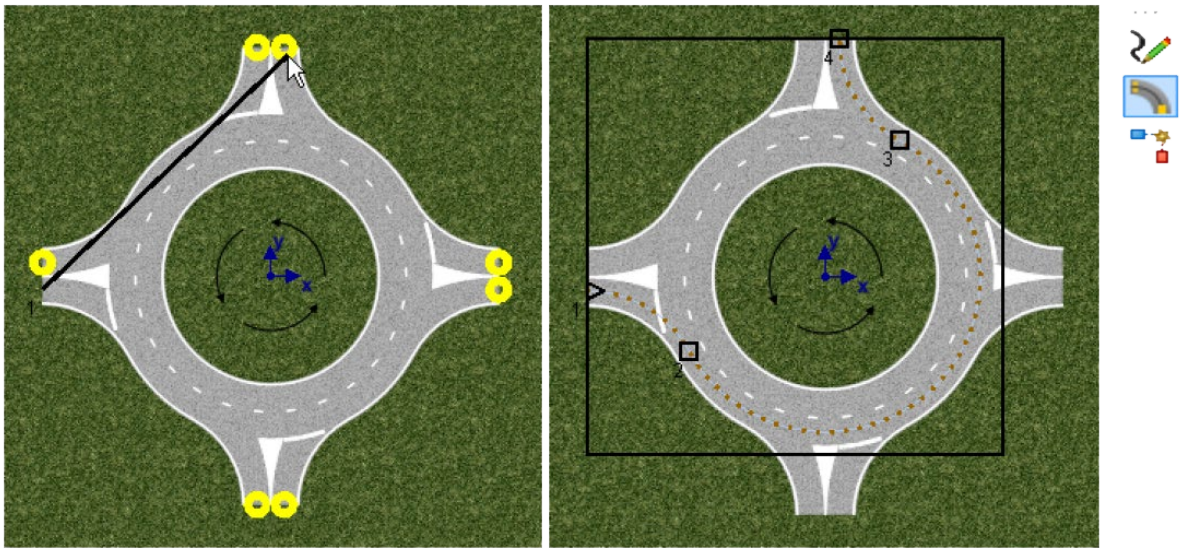
After that we right click on the *Curved Road* segment -> *edit* and then we connect then end of the *Curved Road* segment to the *Roundabout.*



Now before adding an actor, we will first create a path for the actor. For that we have two options a Manual Path Definition and an Inherited Path Definition

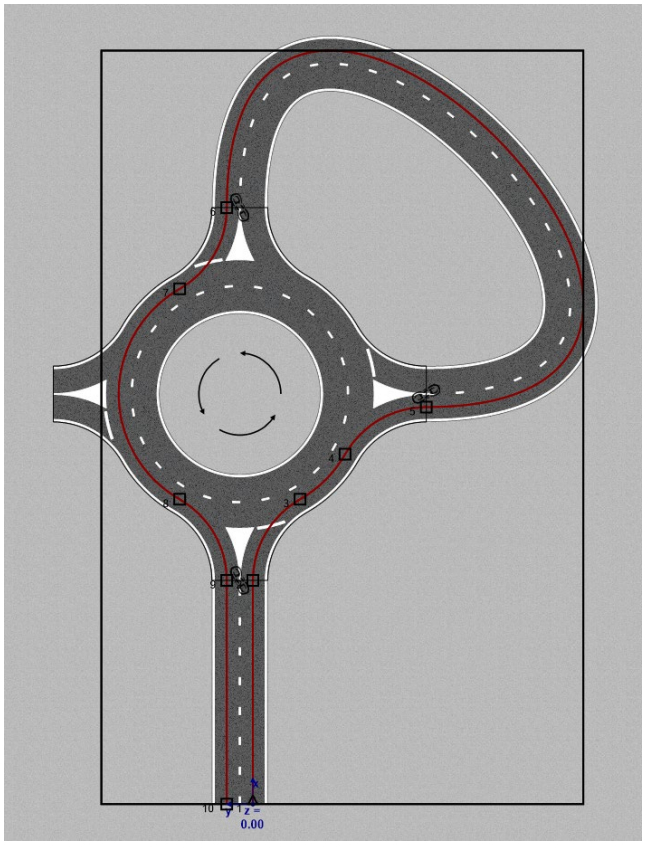


When using the inherited path definition option yellow dots will appear at both ends of each road segment by connecting these dots a path will be created which conforms with the geometry of the road



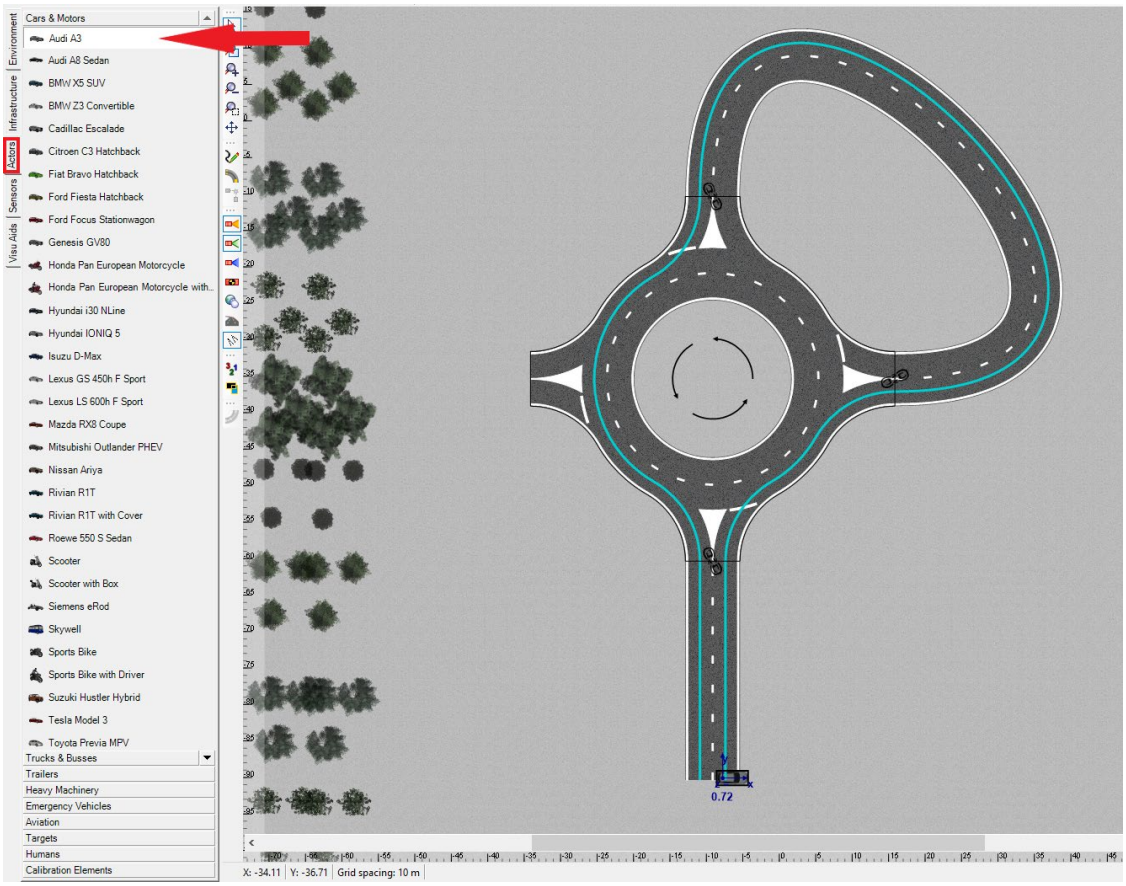
Note: the default setting of an object can be changed in **Settings / Preferences**

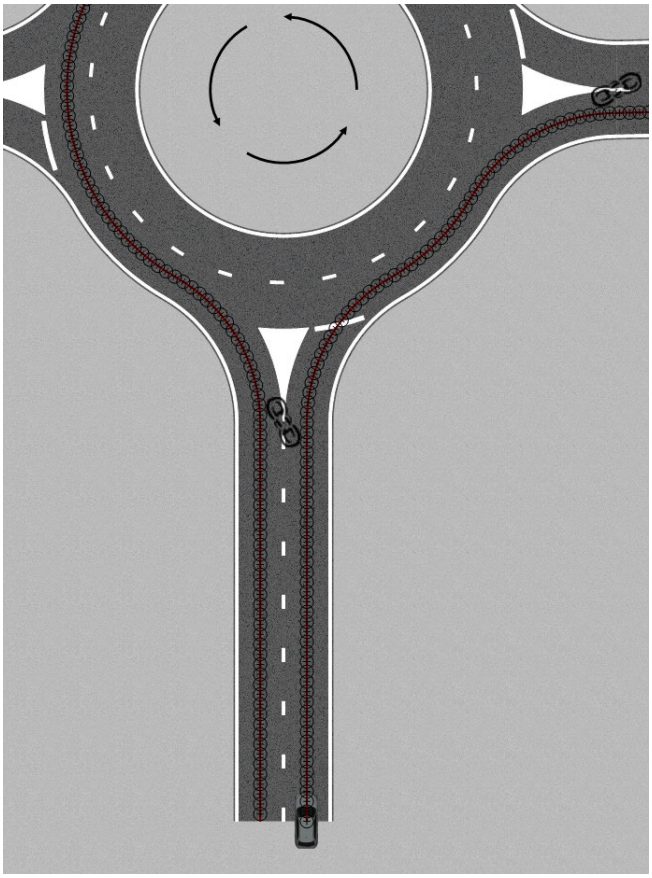
We create our desired path:



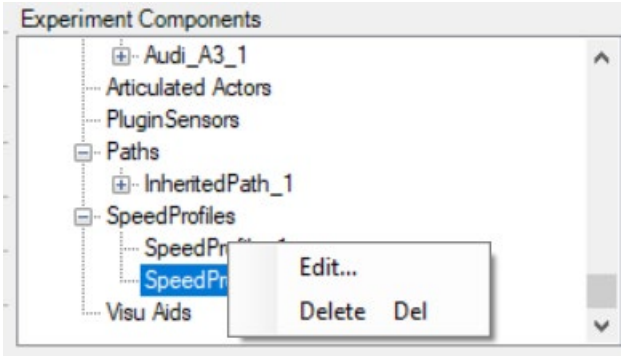
Now we add the actor to our created trajectory, by dragging a car from the *Actors* list and bringing it to the trajectory until the trajectory changes its color to

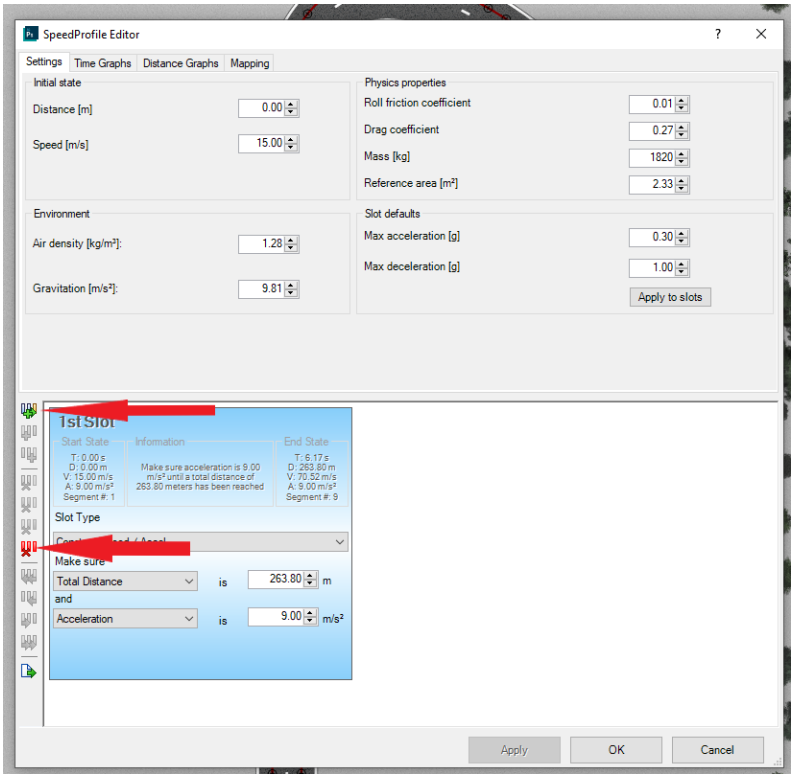
light blue then when we drop the actor we notice that the trajectory line gets small circles over it.



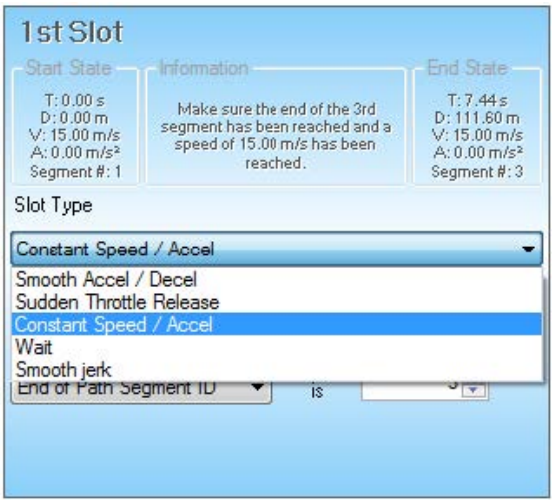


Now we can adjust the speed profile from the *Experiment Components*



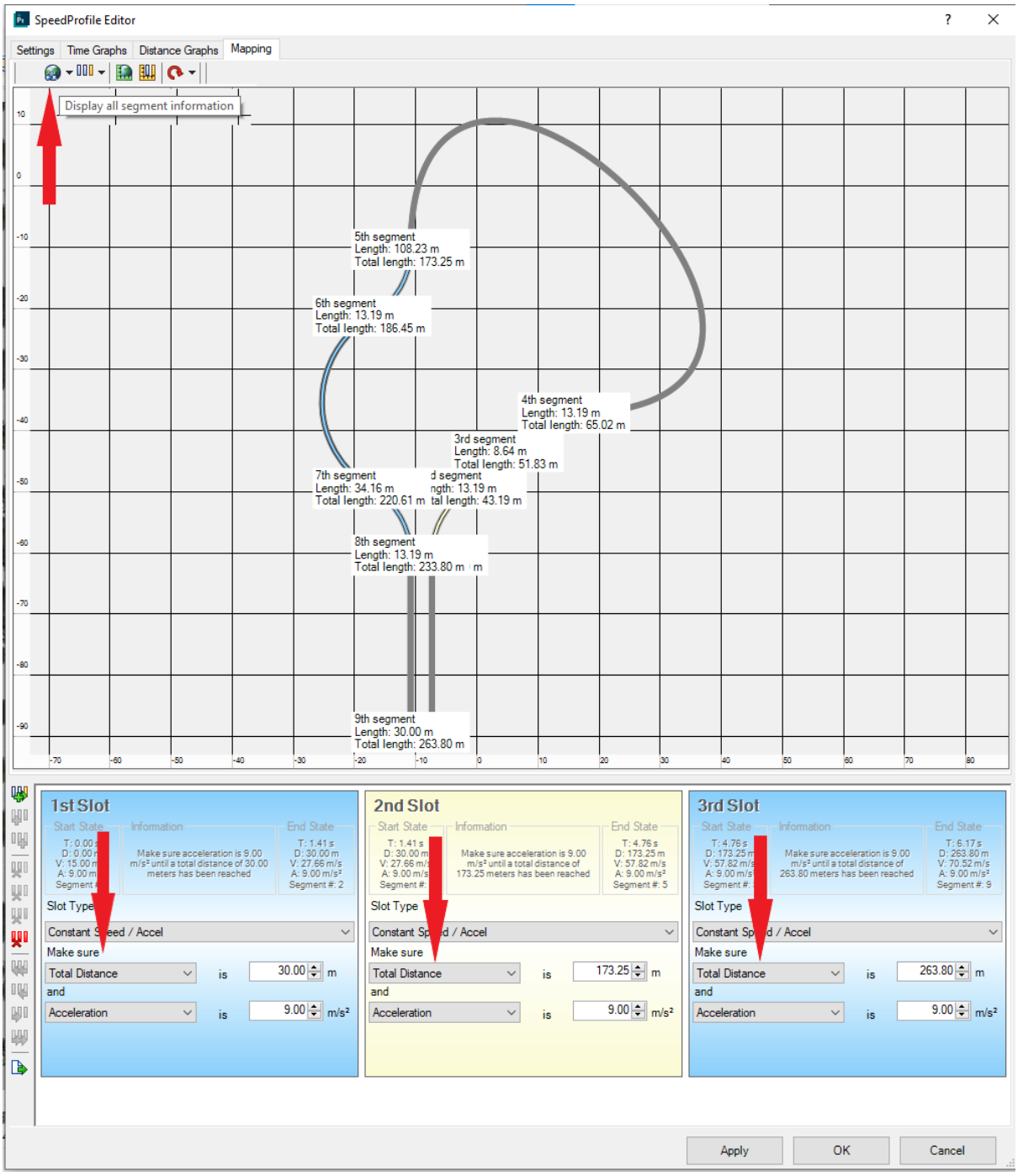


We can add new speed slot using the green plus icon and we delete them using the red cross icon. And we can choose what each speed slot does using the *Slot Type* drop list

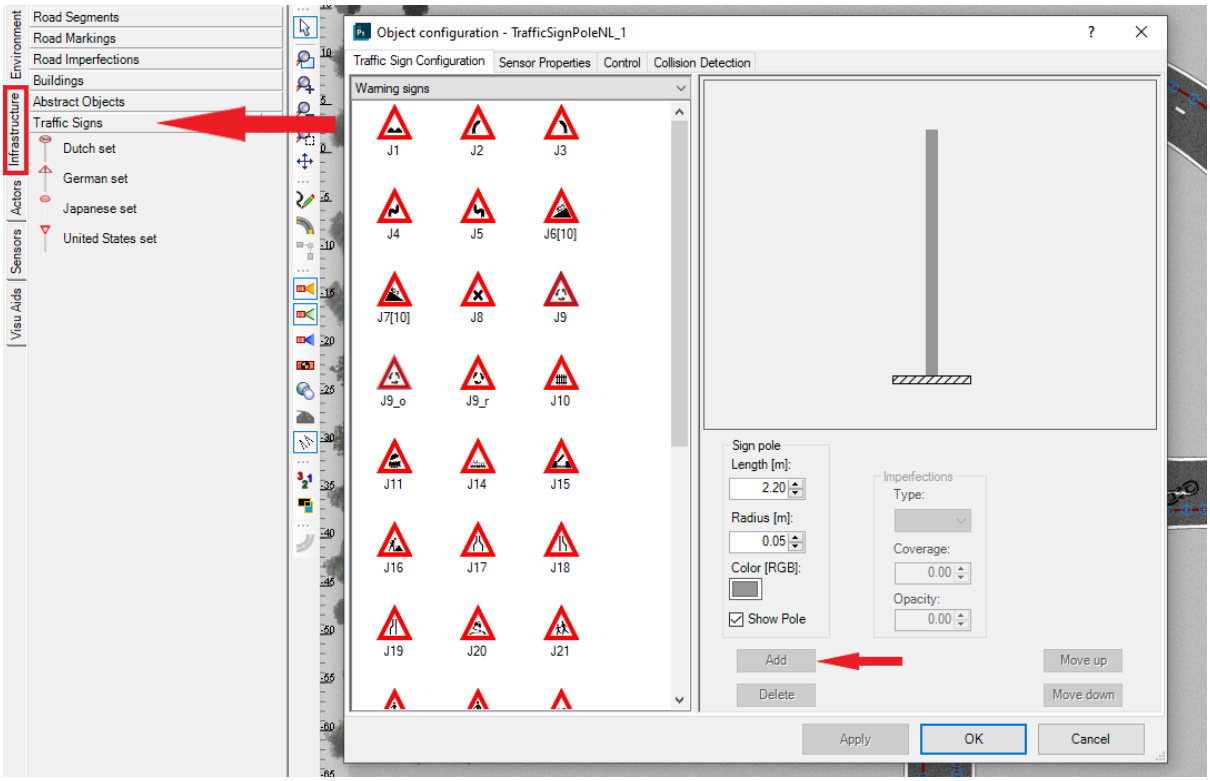


If we go to the *Mapping* tab of the *SpeedProfile* Editor window we can design our specific speed profile easier, for example we will design a speed segments with three speed slots.

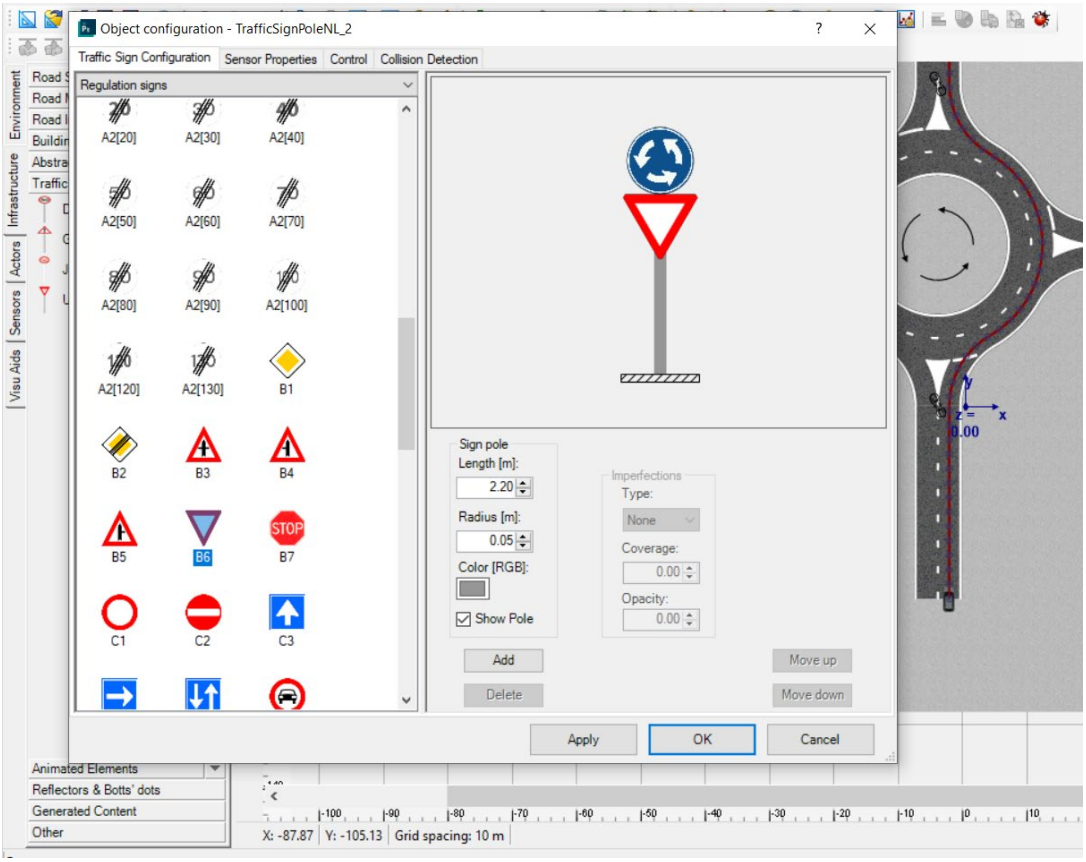
The easiest way is to go to the *Mapping* tab and then activate the "*Display all segment information*" option and then at each speed slot we choose "*Total Distance*" from the "*Make sure*" drop list options, and then for each slot we read the *Total length* data of each segment and we create the speed profile accordingly.



Now from the *Infrastructure* tab we will add a traffic sign from the *Traffic Sign* list, we drag and drop it to our build area and then the below windows opens.

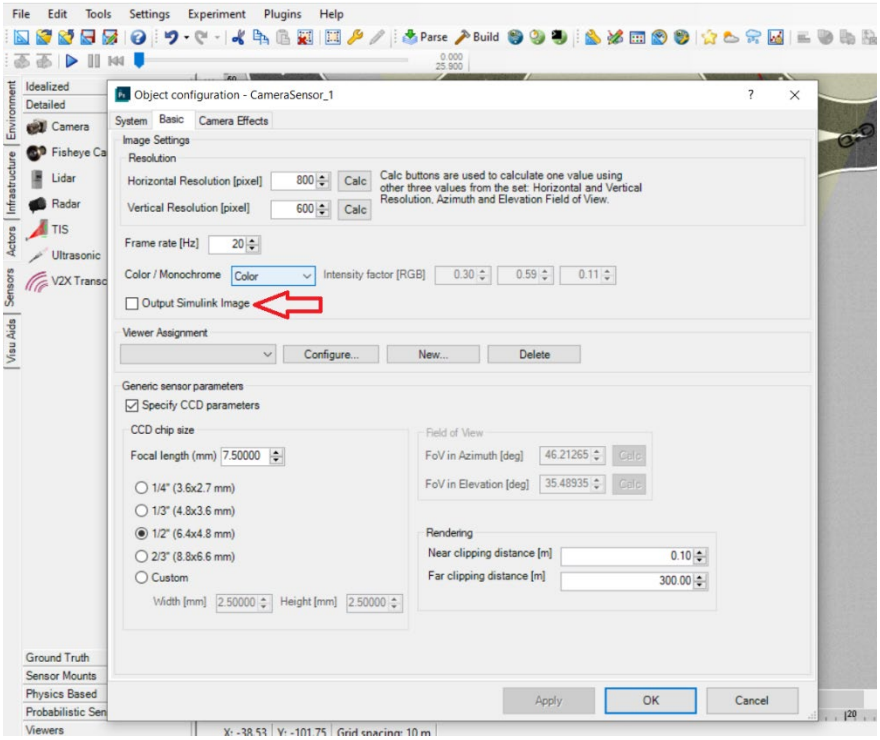


We choose the sign, and we double click on the chosen sign or we click add

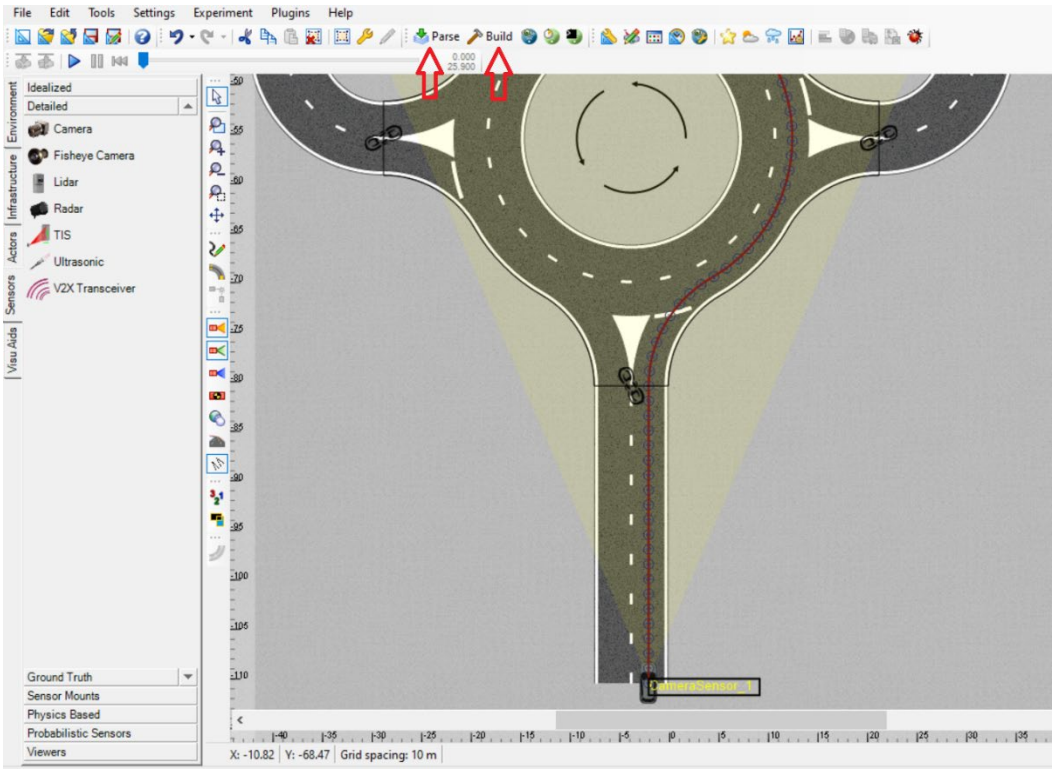


Finally we will add a camera on the car to have a view of the actor, for that we go to the sensor tab and then detailed and from the list we choose camera and we drag and drop it on the car the object configuration window will open and from

there we go to the Basic tab and we check the box next to the Output Simulink Image option.

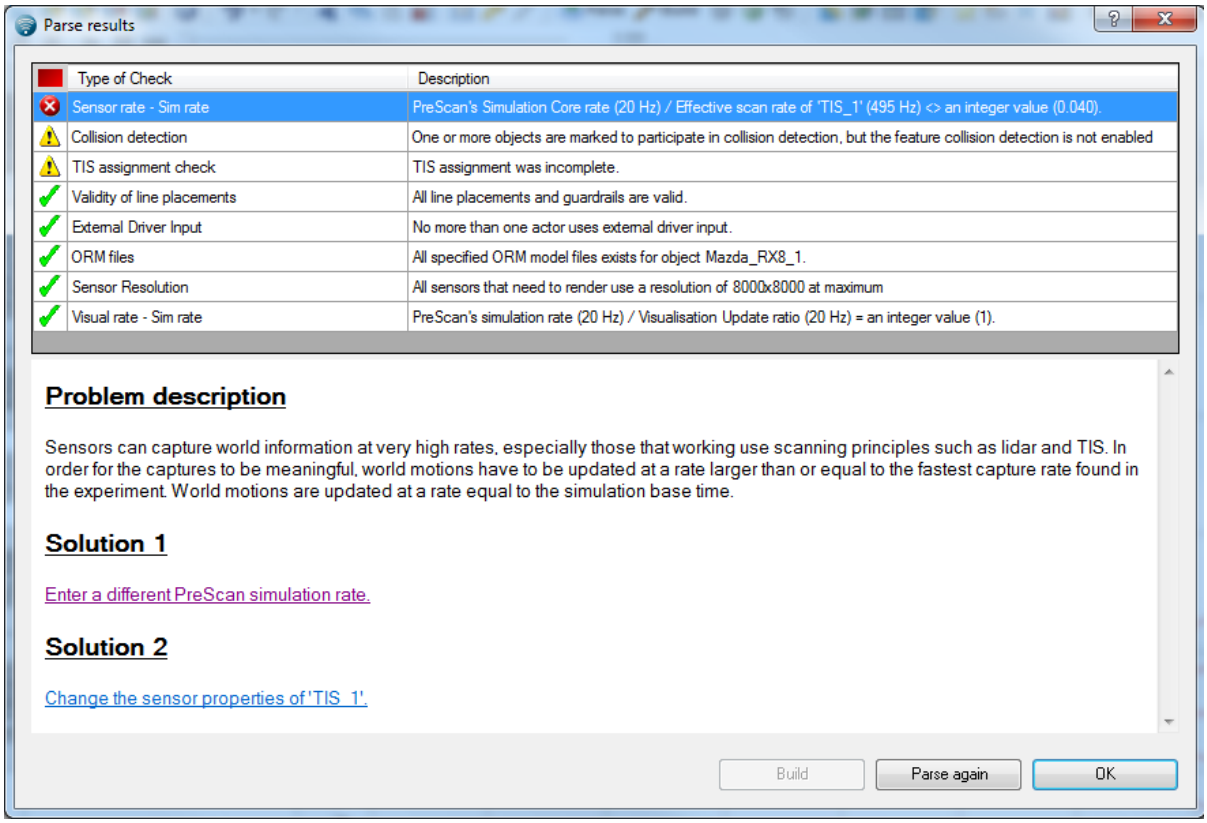


Now our simple experiment is ready, now we go to the toolbar, and we click *Parse* and then and *Build* to create the Simulink file for the experiment

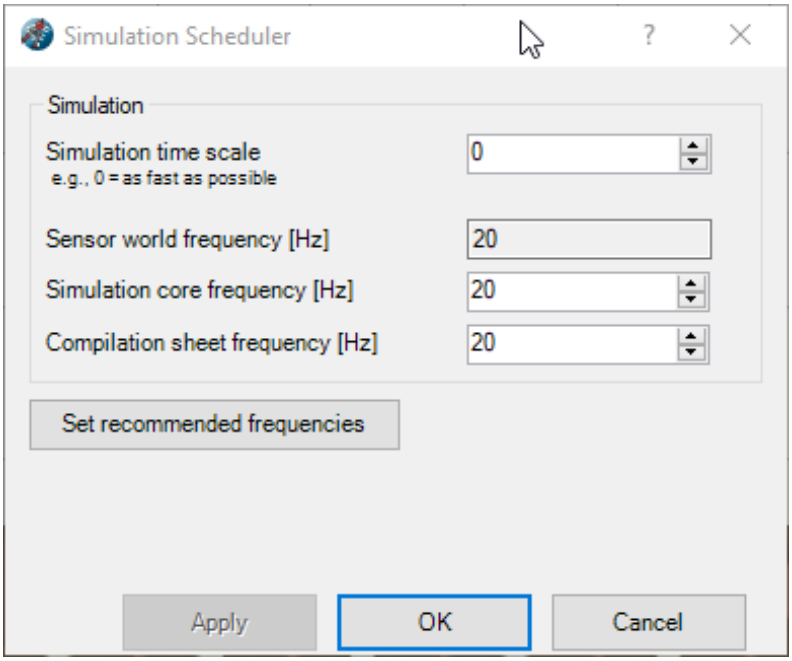


Parse & Build Mechanism: PreScan consolidates all warning dialogs into a Parse & Build module which checks for simulation rate consistencies. It allows you to

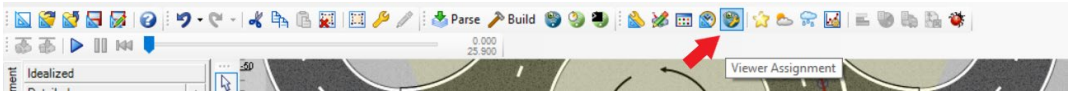
correct any identified problem areas and build your experiment by generating necessary files for 3D visualization and running the experiment.



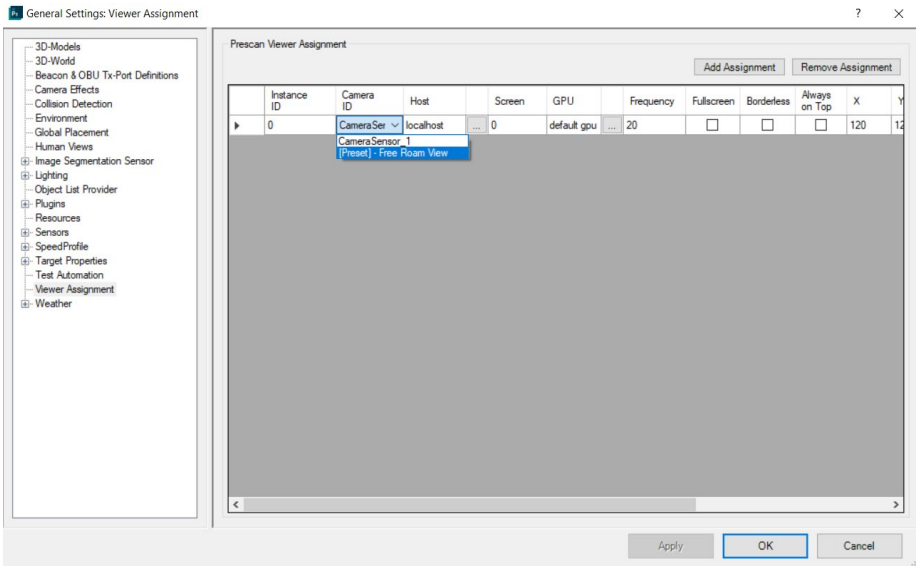
These functionalities, along with PreScan's scheduler for managing simulation rates, make PreScan a robust tool for creating and managing complex virtual environments.



One last step is to add a world view to better visualize our experiment we can do that by going the tool bar and clicking on the Viewer Assignment

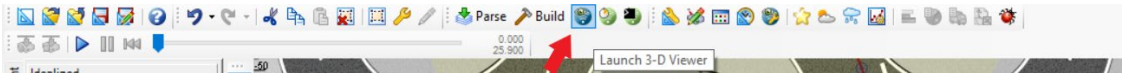


The below window will open.

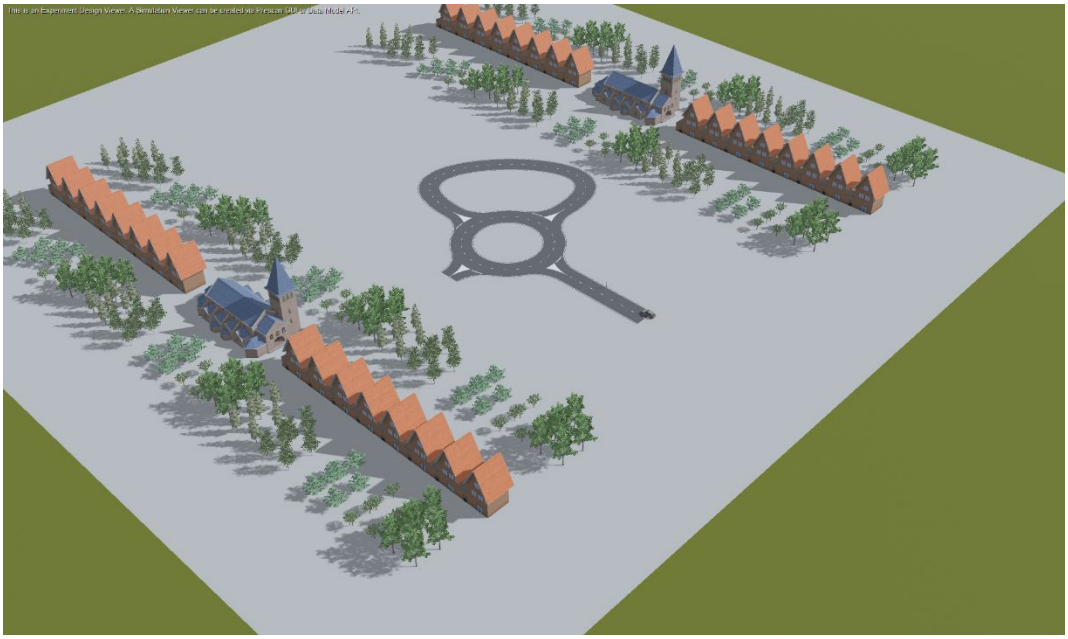


We click on Add Assignment, and from the Camera ID drop down we choose Free Roam View.

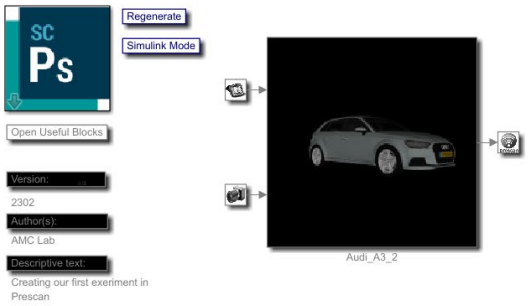
Before we start the experiment, we can use the built-in viewer from Prescan to ensure that the scenario looks like what we want.



In our case it looks as below:

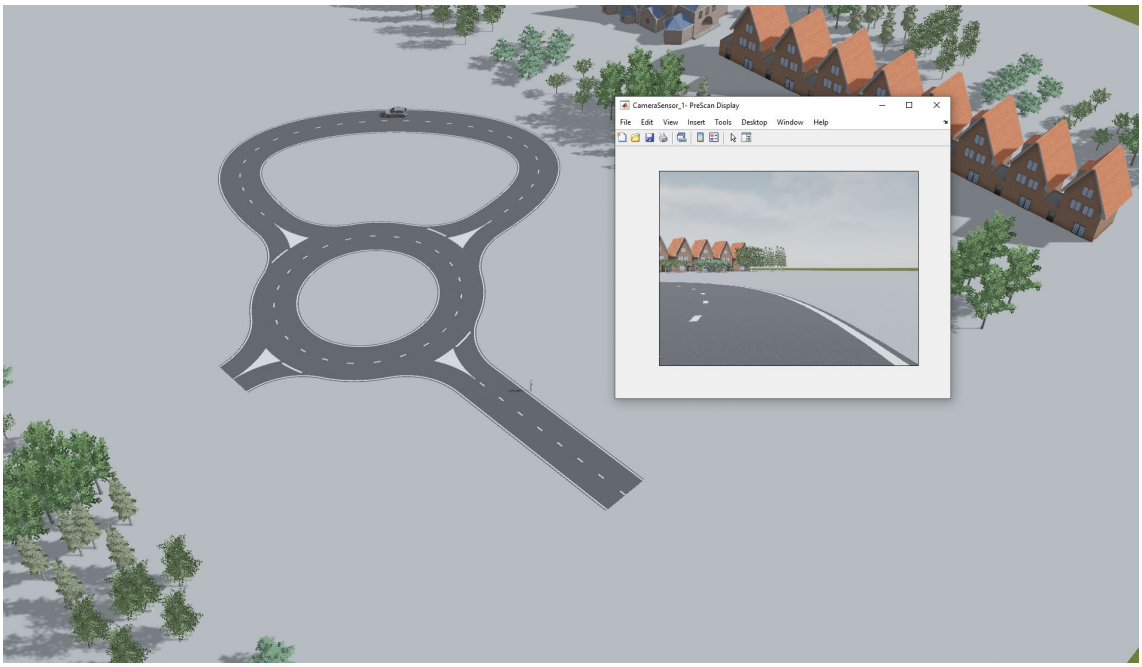


In order to run the experiment, we go to the directory of our experiment (D:\PreScan2302_Experiments\Test_Experiment) and we go to the Simulink file that has the same name of our experiment.

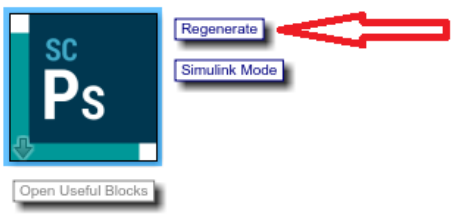


Num ID	OBU-Beacon ID	Type ID	Type Tag	SD	CD	Name
383		1	Car	Y	N	Audi_A3_2

By running the Simulink file, the experiment shall run smoothly



If we apply any changes to the created experiment in Prescan in order for this changes to be applied in Simulink we first need to rebuild the experiment in Prescan and ensure that it is built without errors. Afterwards, we need to update by pressing Ctrl+D or selecting Update Diagram from the Simulation menu, if the model updates without any errors only then we are able to regenerate the experiment by clicking on the regenerate icon in the experiment Simulink file



Appendix

What is Prescan

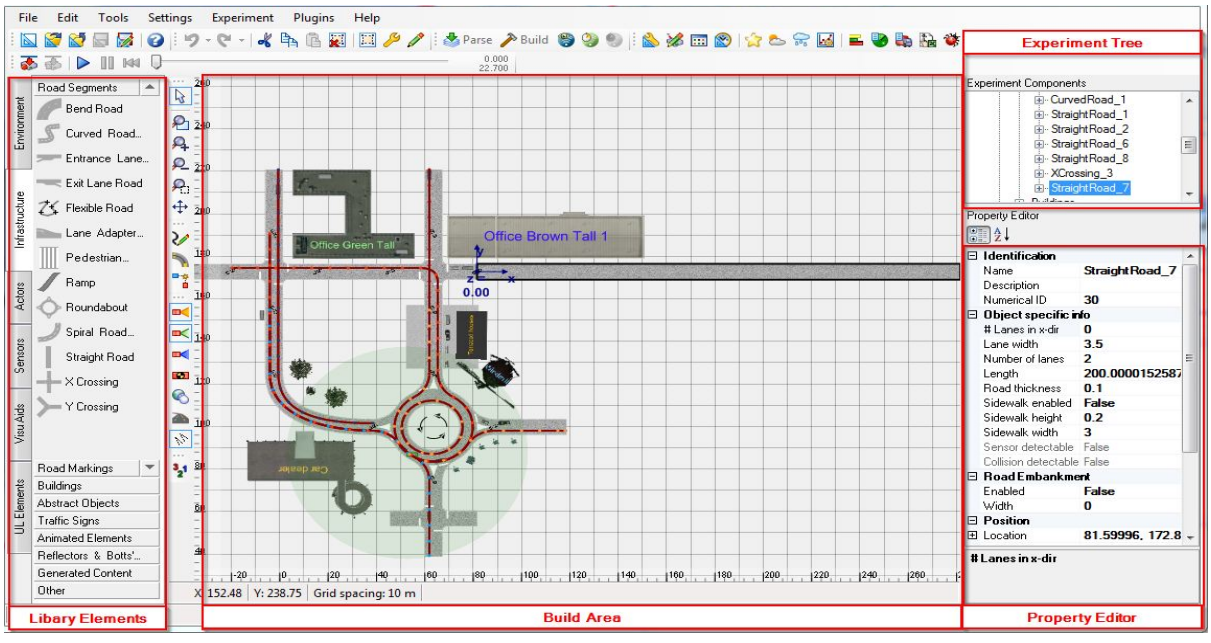
PreScan is a comprehensive tool designed for the development and evaluation of sensor-based road vehicle safety systems. It simulates real-world scenarios and uses physical sensor models to help vehicles respond optimally to their surroundings.

The tool, fully integrated with MATLAB/Simulink, allows for seamless collaboration across technologies and disciplines. The base version includes a powerful graphics pre-processor, a 3D visualization viewer, and a MATLAB/Simulink connection, making it instrumental for addressing challenges in sensor data interpretation, system response optimization, and managing unexpected disruptions.

Prescan GUI

The GUI of Prescan is intuitive and user-friendly, equipped with a library of predefined and customizable elements. These allow for quick experiment building via drag & drop actions. It also features robust reporting, preview, and parsing mechanisms, providing insights into the constructed experiment and helping identify any discrepancies that need rectification before execution in the Engineering Workspace. The GUI also takes care of automatic conversion of older experiments, easing the transition between different versions.

In Prescan's GUI we have the sections shown in the figure below:



The PreScan interface is designed for ease of use and efficient navigation:

1. Left-hand side: Here, you'll find tabs representing a variety of library elements, including actors (like cars, animated humans, trucks, etc.), infrastructural components (like buildings, roads, trees, etc.), and various sensor models.

2. Upper-right section: This area features an 'experiment tree' which showcases the relationships between elements in your experiment. You can quickly determine the type of sensor mounted on a vehicle and its assigned trajectories. Additional information that may not be directly visible or needed in the 'property editor' (situated just below the experiment tree) can be accessed through the 'object configuration dialog box'. This is invoked by right clicking the selected object.

3. Lower-right section: This is the 'property editor' that complements the experiment tree.

4. Center: This is your 'build area' providing a top-down view of all elements placed in PreScan's world. This area serves as the primary user interface for setting up your experiment.

Keep in mind, options such as copy, paste, and edit can be accessed by right clicking either on the object in the experiment tree or directly in the build area.

PreScan offers a comprehensive library divided into five categories:

1. Environment: This includes underlays.

2. Infrastructure: This comprises road segments, buildings, scalable abstract objects, nature elements, and traffic signs.

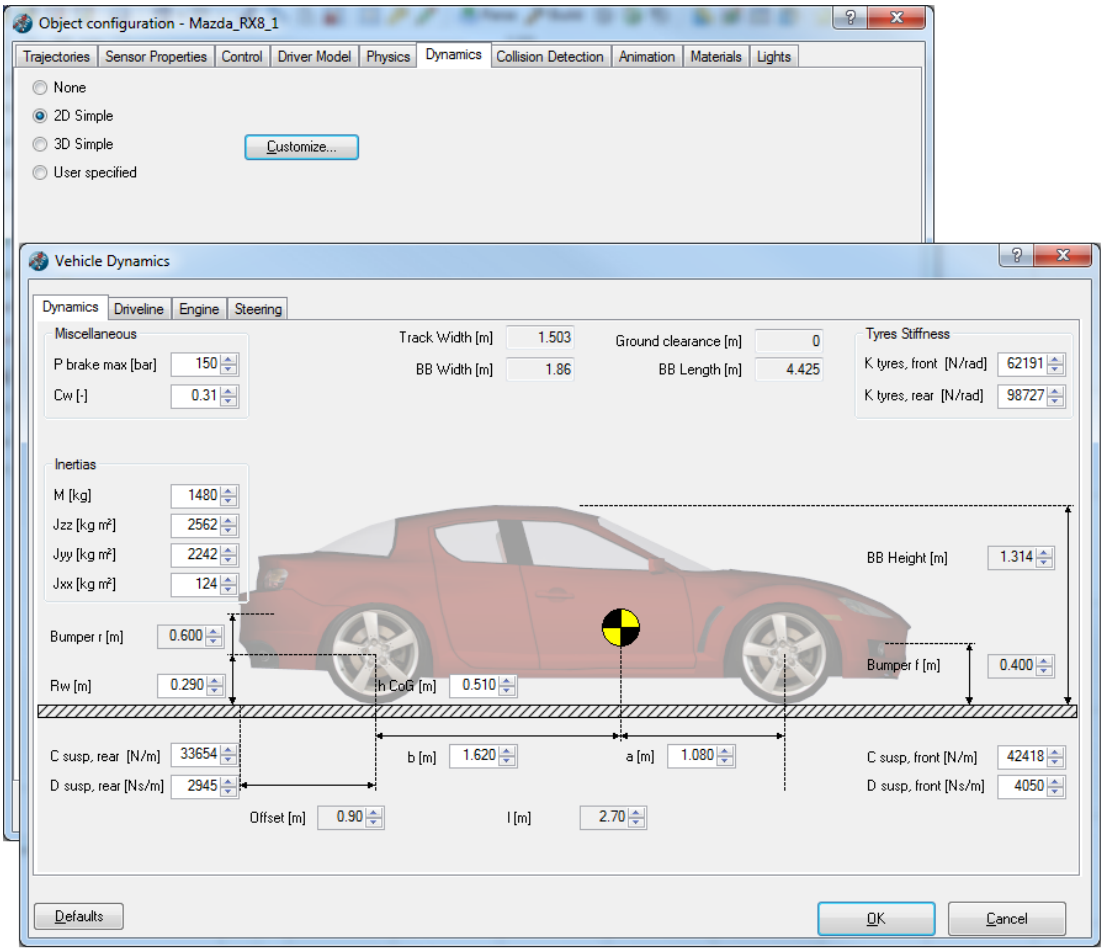
3. Actors: This involves cars, trucks, buses, humans, and calibration elements for sensor readings.

4. Sensors: This includes idealized and detailed sensors, and sensor mounts.

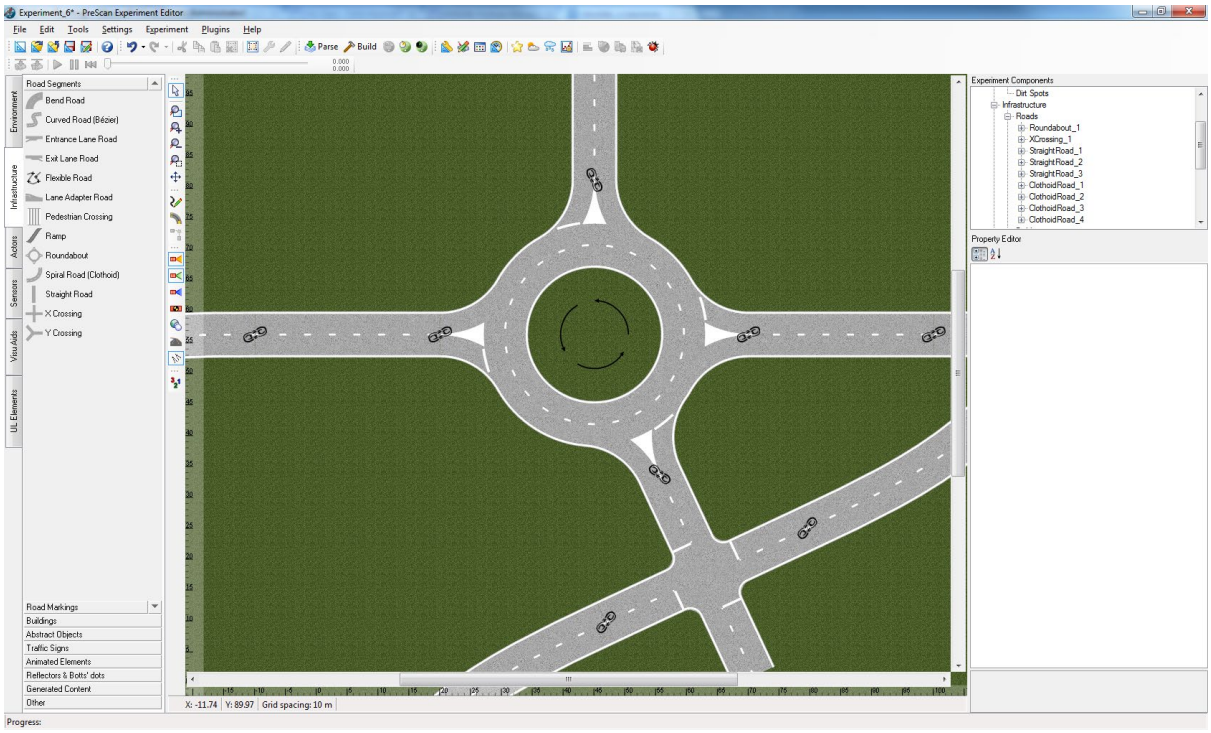
5. VisuAids: This offers a predefined human view that can be placed onto a car or anywhere in the world.

Let's explore some of these elements:

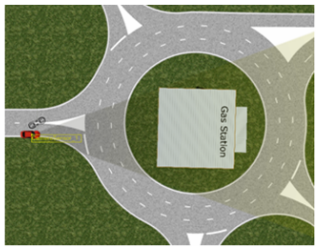
Actors: These are the active elements in your experiment, like cars and pedestrians. You can assign them sensor properties, dynamic models, control models, and animation properties. Actors can be linked to follow certain trajectories. There are also specialized actors such as pedestrians, cyclists, and motorcyclists.



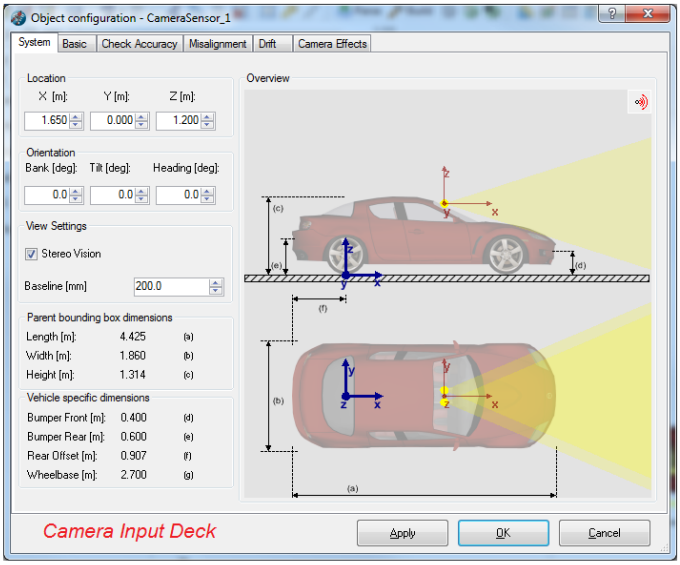
Road Segments: Found under the Infrastructure library, road segments include standard and special variants like entrance and exit lanes, merge segments, and curved roads. They're fully customizable, allowing you to set lane numbers, driving directions, line types, colors, and more.



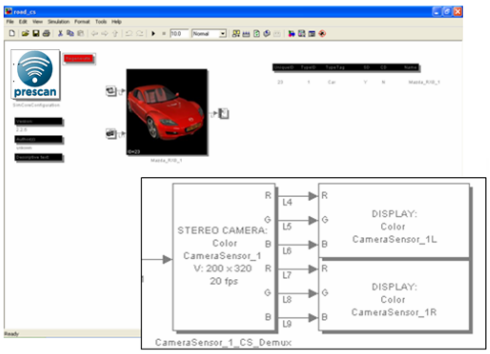
Camera Sensor: You can place and configure vision sensors on actors, enabling them to output camera images to the run-time environment.

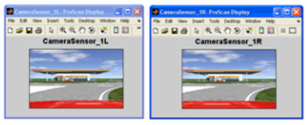


Situation Overview



Camera Input Deck

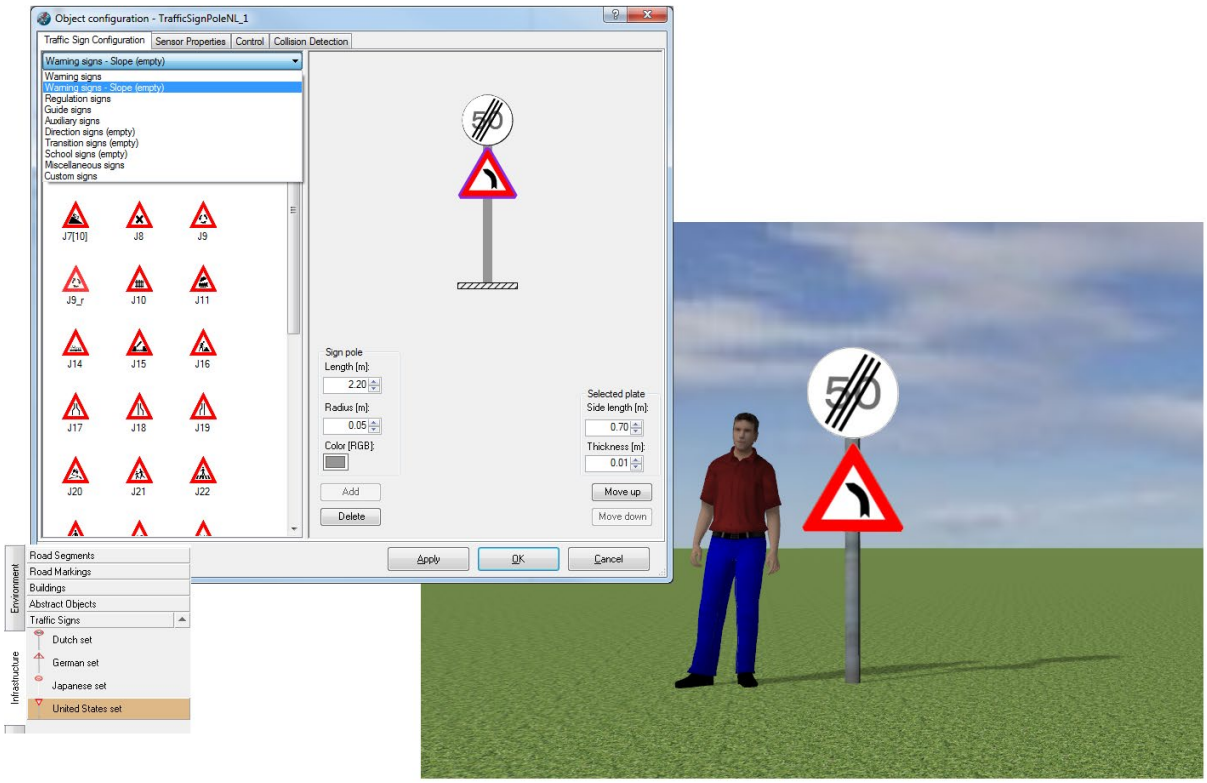




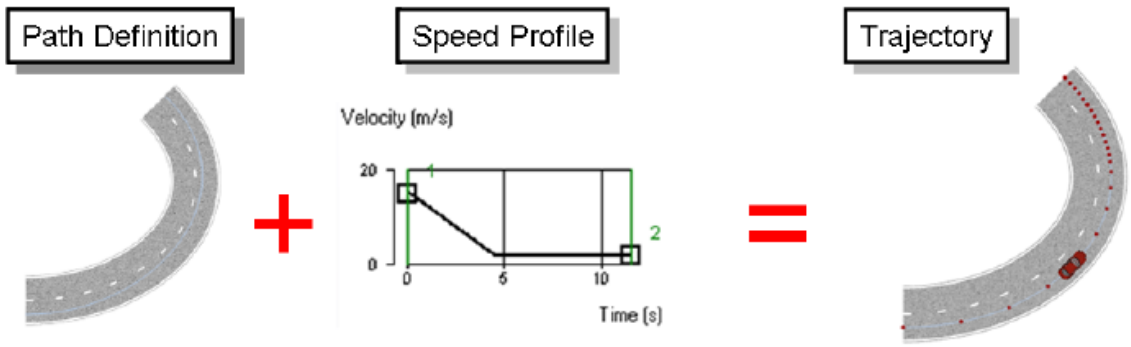
Compilation Sheet Representation

Buildings and Abstract Objects: These elements, found under the *Infrastructure* library, can be used to set up realistic scenes. Abstract objects are useful for scene decoration without significant emphasis on textures. Both types of objects can have sensor properties assigned.

Traffic Signs: You can rapidly create custom traffic signs within the Infrastructure library.



Car Maneuver Definition: Car maneuvers can be defined using the GUI's specialized tools. You can construct paths by combining segments and assign these paths to actors to create trajectories. Speed profiles can be edited, and paths modified using the path editor. You can also synchronize actor positions to simulate near-crash situations.



Separate paths combined with separate speed profiles result in trajectories