

Starting with Scenic and Sumo Integration

To start with the integration of SCENIC and SUMO, we started with downloading the software/library for both. We encountered some issues with downloading SCENIC such as it not working with Python 3.9 even though it said it worked with 3.7 or newer. To solve this, we downloaded 3.8 instead, which ended up solving our problems.

We ran some of the example scenarios provided in the examples folder to get a general understanding of the capability of SCENIC. From there we moved to the CARLA model.scenic file to start creating one for SUMO. Carla's simple model.scenic made it easy to get an idea of what is contained in a model.scenic file and what its purpose is. After some time, we created several scenes to show the capability of the current Scenic Sumo integration. Next is to work on implementing the distribution functions in the scenes.

What is SUMO

Simulation of Urban Mobility is used by transportation professionals. It's great for simulating a large number of actors and/or for simulating scenes quickly. Since it is low fidelity, it can be run quickly and it has an option to run without a graphical user interface which speeds up time even further.

Running a Scene

To create a scene the user must download our repository off of GitHub which can be found here: <https://github.com/AkbasLab/scenic-sumo/tree/main> . After downloading the repository create a folder called "sumo" in the *scenic/simulators* folder. Copy the *model.scenic* file that is found in the repository to the new *sumo* folder you just created. To download SUMO, follow this link: <https://sumo.dlr.de/docs/Downloads.php> .

The next step is to open the scenario folder and click on one of the scenes below. The scenic file should specify which *.sumocfg file it needs to run the simulation commented at the top of the file. In the *map* folder there is a list of folders and one of which should match the name of the *.sumocfg file that you read in the scene. Open Netedit and open the *.net.xml version of the *.sumocfg file. From here a road network should fill the screen. Click the *Edit* tab and then click "Open in sumo-gui". Save this file with the *.net.xml file and make sure the folder, *.net.xml, and *.sumocfg file all have the same name. The purpose for resaving the *.sumocfg file is to set the correct path to the xml file in the *.sumocfg file. This will need to be done for each map that you use.

Now you should be able to run a scene by running the following command while in the folder:

```
python scenic-sumo.py -sc filename.scenic -so filename.sumocfg
```

Other Files

The excel file that this is paired with this is the template from which these scenarios were created.

Scene 1S1.scenic -

The first scene has two vehicles that drive along a curved road with two lanes. The driver under test (DUT) drives in the left lane along the road. The red vehicle accelerates from a starting point behind the DUT in the right lane. After reaching a point at which the red vehicle can merge in front of the DUT it will do so. This is the end of this scenario.

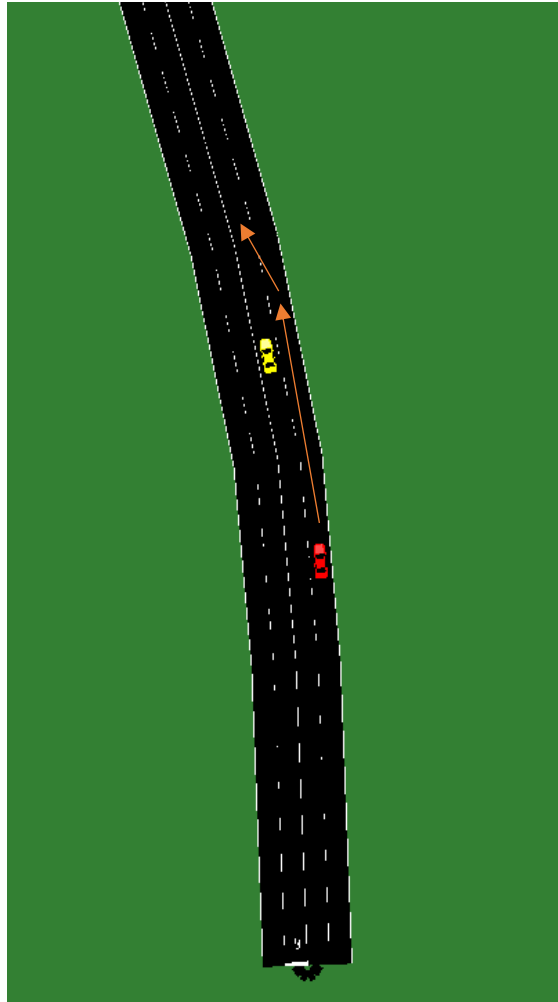


Figure 1. Shows the path of the vehicles in the 1S1 scene. The road network is CurvyRoad.

Scene 1S7.scenic -

The second scene also has two vehicles along a curved road with two lanes. The DUT is in the right lane and the red vehicle is in the same lane just behind the DUT. The red vehicle is accelerating behind the DUT. From here the red vehicle will either pass the DUT or it will rear end the DUT depending on the specified constraints given by the user. The scenario ends when the vehicle passes the DUT.

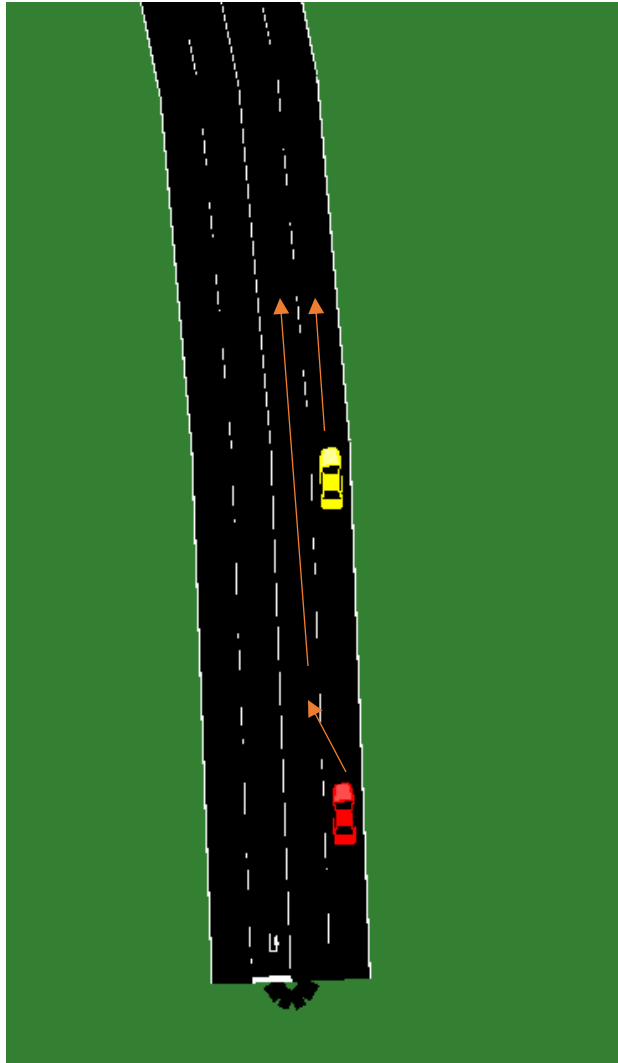


Figure 2. Shows the path of the vehicles in the 1S7 scene. The road network is CurvyRoad.

Scene bl1s18.scenic -

The third scene is two vehicles on the same curved road with two lanes. The DUT is in the right lane and the red vehicle is ahead of and to the left of the DUT. The red vehicle changes lanes into the DUT's lane. The scenario ends after the vehicle changes lanes into the DUT's lane.

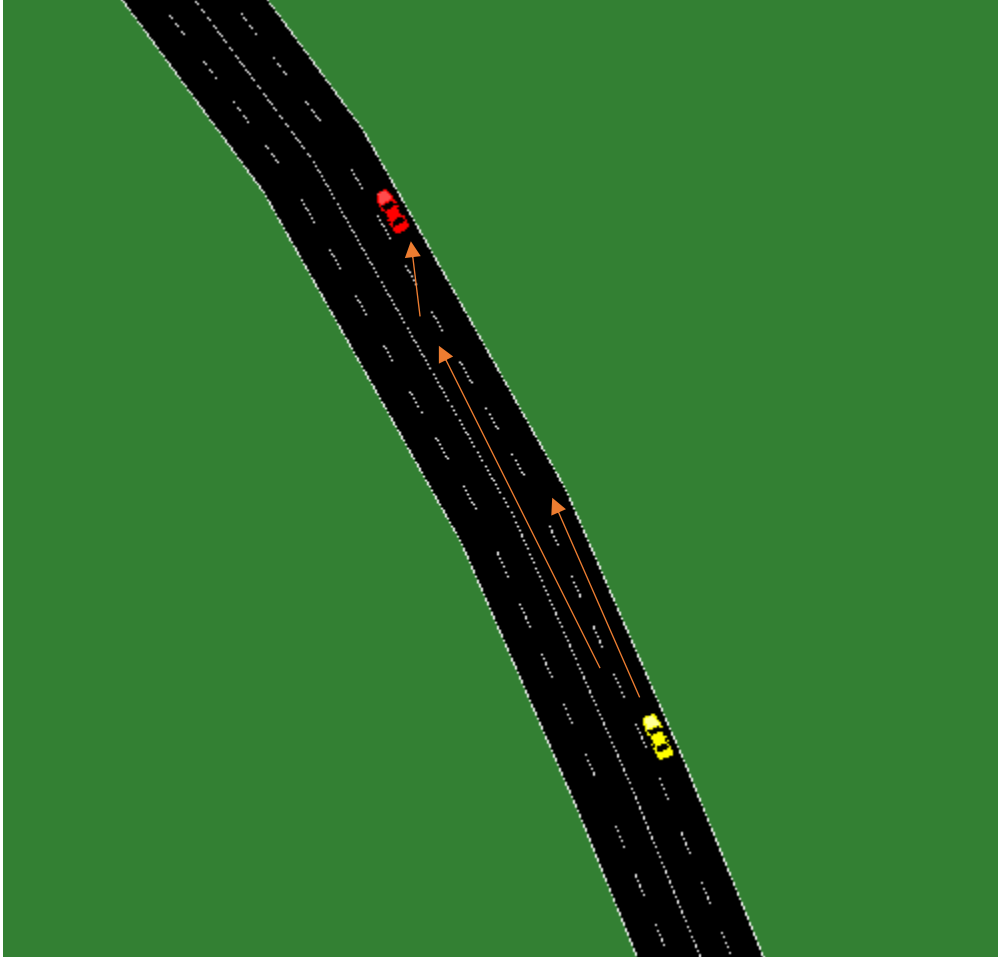


Figure 3. Shows the paths of the vehicles in bl1s18 scene. The road network is CurvyRoad.

Scene 2S13.scenic

The fourth scene has three vehicles in the same road. The DUT is in the right lane and is the red vehicle. Vehicle one is in the left lane and vehicle two is parked in the right lane some distance up the road. Vehicle one maintains its speed and continues straight while the DUT must slow down and merge into the left lane to get around vehicle two. Once the DUT is past the parked vehicle the scenario is over.

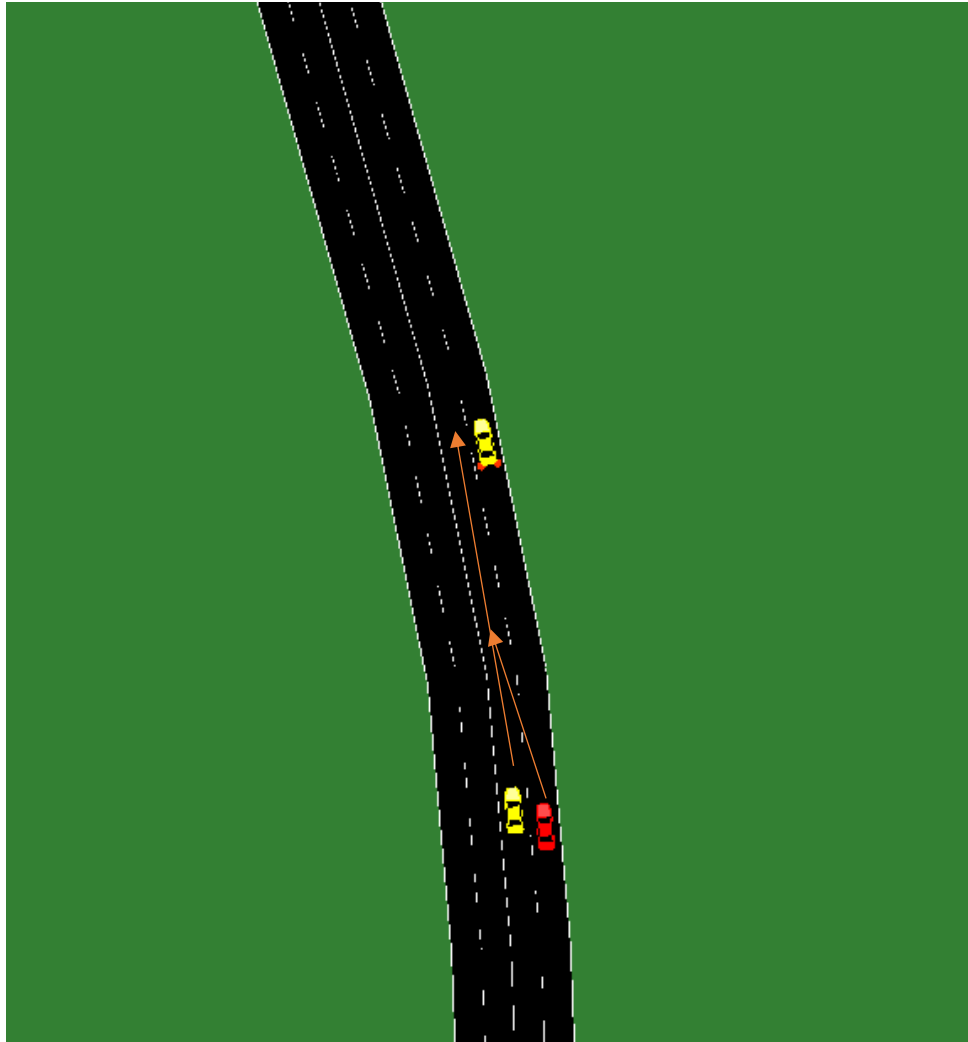


Figure 4. Shows the path of the vehicles in the 2S13 scene. The road network is CurvyRoad.

Scene 1X.scenic -

The fifth scene occurs at a four-way intersection. The DUT is approaching the intersection from any edge. The other vehicle enters from the left or right edge and runs the red light. Neither vehicle breaks in this simulation. The scenario ends when the two vehicles exit the intersection.

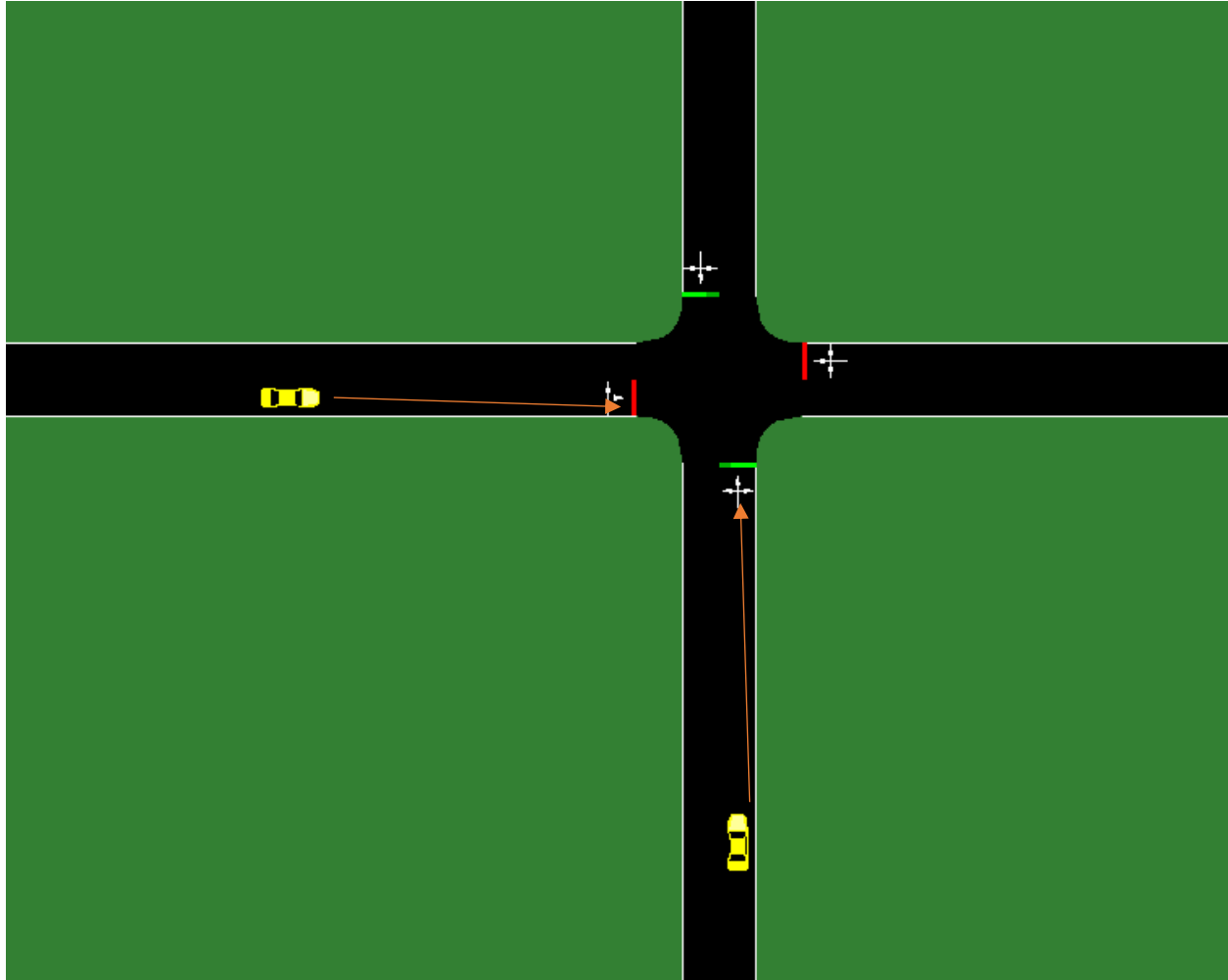


Figure 5. Shows the path of the vehicles in the scene. The road network is 1X.

Scene 1T1.scenic -

The last scene occurs at a three-way intersection. The DUT is approaching the intersection from the bottom and the other vehicle is approaching from the right. The other vehicle is braking as it enters the intersection. The other vehicle runs the red light in this scene as well. The scene ends as the vehicles exit the intersection.

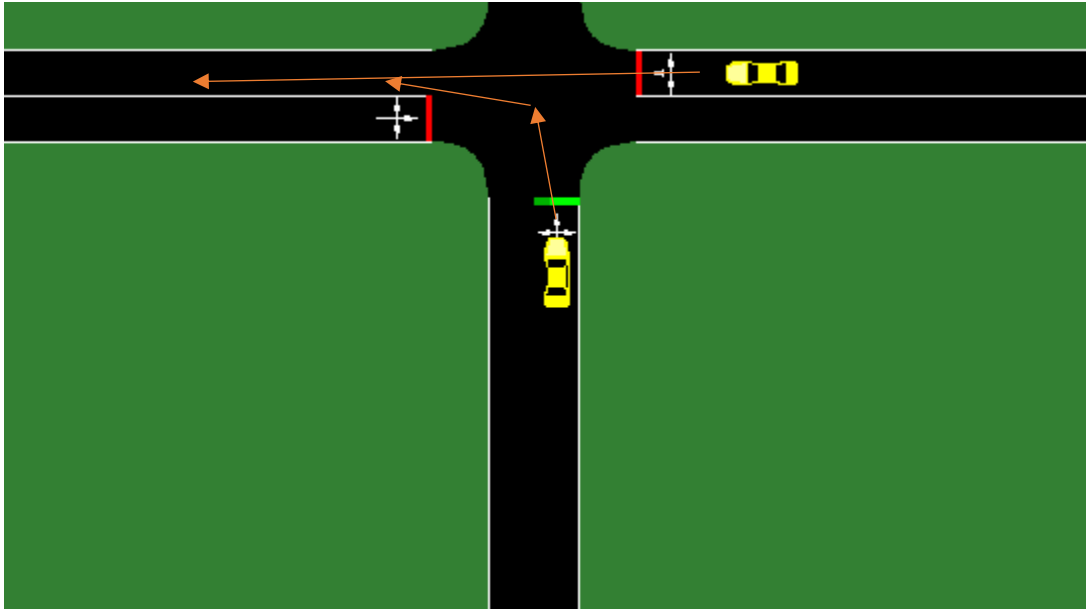


Figure 6. Shows the vehicle path for the 1T1 scene. The road network is 1X.