

Mapping Group Report

Dec. 12, 2021

Introduction

Set up process

Getting the map

Editing the map

Installing, Downloading, and Running

Python

CARLA

RoadRunner

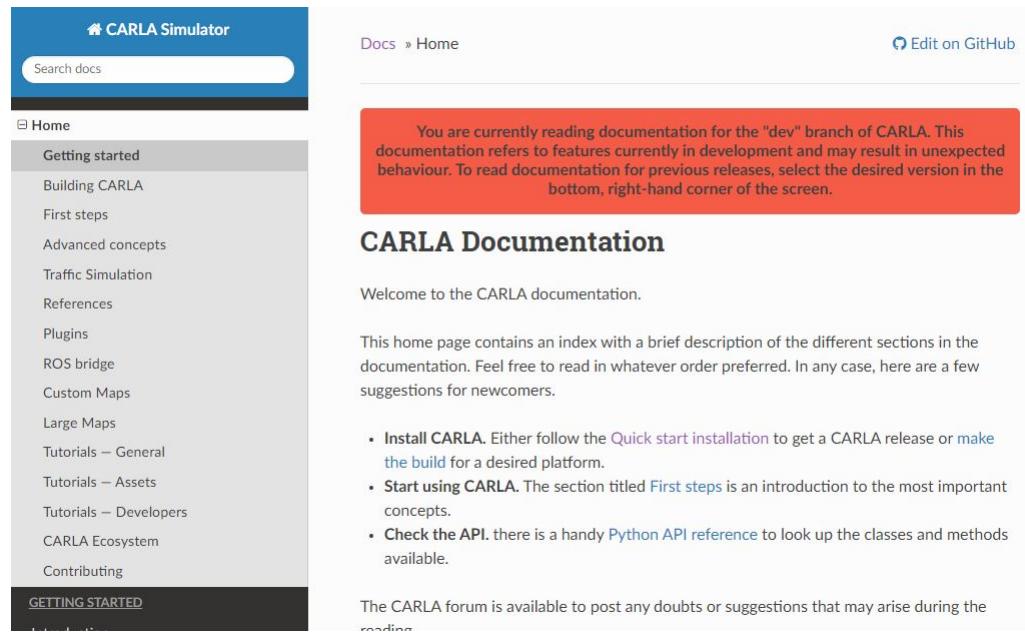
CARLA

- CARLA is an open-source autonomous driving simulator. It was built from scratch to serve as a modular and flexible API to address a range of tasks involved in the problem of autonomous driving.



Installing and Running Carla Simulator

- **Hardware and Software Requirements**
 - System Requirement:
Windows/Linux
 - An adequate GPU
 - ~20 GB Disk Space
 - Python
 - Python 3.7 is recommended
- **Detailed Guide:**
<https://carla.readthedocs.io/en/latest/>
 - Also available in our google drive



The screenshot shows the official CARLA documentation website at <https://carla.readthedocs.io/en/latest/>. The page has a blue header bar with the title "CARLA Simulator" and a search bar. Below the header is a sidebar with a "Home" button and a list of links under "Getting started": Building CARLA, First steps, Advanced concepts, Traffic Simulation, References, Plugins, ROS bridge, Custom Maps, Large Maps, Tutorials – General, Tutorials – Assets, Tutorials – Developers, CARLA Ecosystem, and Contributing. At the bottom of the sidebar is a "GETTING STARTED" button. The main content area has a "Docs » Home" breadcrumb and an "Edit on GitHub" link. A red callout box in the top right corner states: "You are currently reading documentation for the "dev" branch of CARLA. This documentation refers to features currently in development and may result in unexpected behaviour. To read documentation for previous releases, select the desired version in the bottom, right-hand corner of the screen." The main content starts with "CARLA Documentation" and "Welcome to the CARLA documentation." It then provides suggestions for newcomers and lists several bullet points: "Install CARLA. Either follow the [Quick start installation](#) to get a CARLA release or [make the build](#) for a desired platform.", "Start using CARLA. The section titled [First steps](#) is an introduction to the most important concepts.", and "Check the [API](#). there is a handy [Python API reference](#) to look up the classes and methods available." At the bottom, it mentions the CARLA forum.

RoadRunner

- RoadRunner is an interactive editor that lets you design 3D scenes for simulating and testing automated driving system.
- You can customize roadway scenes by creating region-specific road signs and markings.
- In this semester, we primarily used RoadRunner to build Park Street

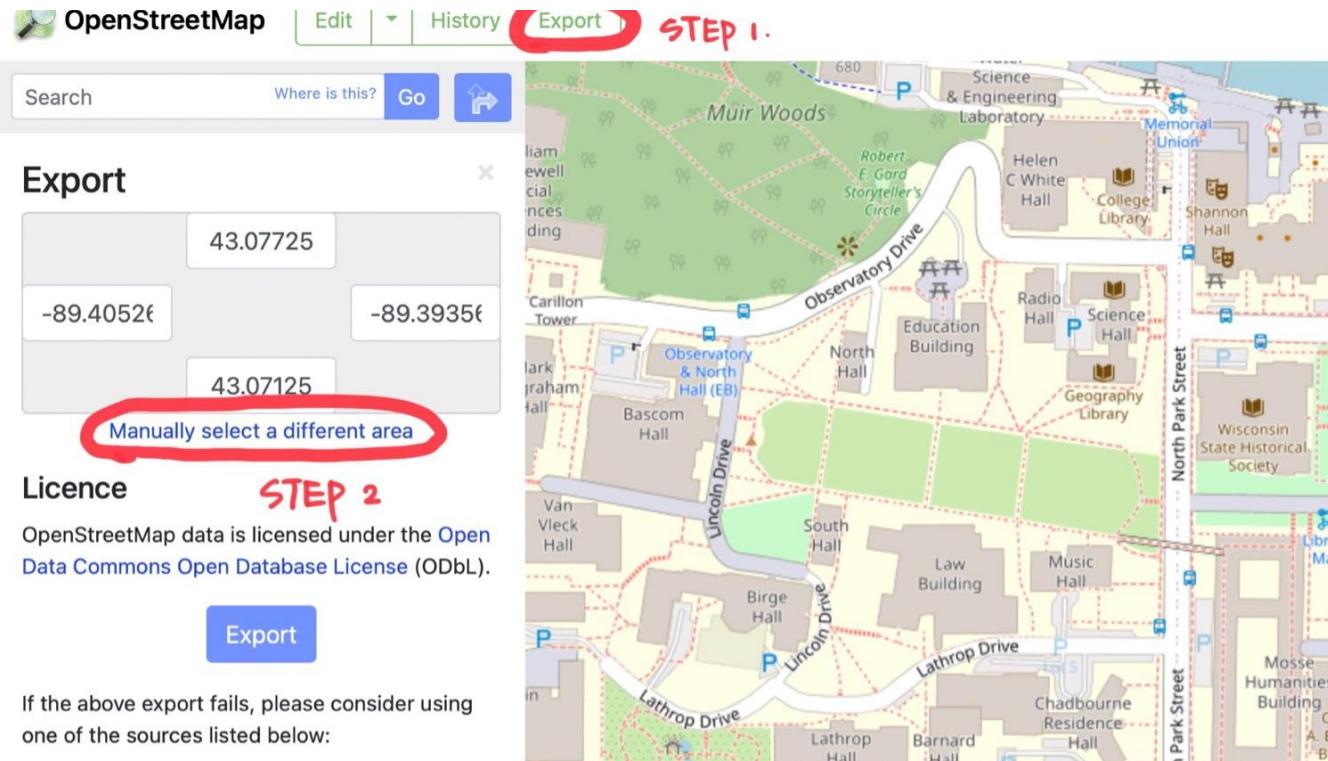


Installing RoadRunner

- Sign in to your MathWork Account with you wisc email
- Send Email to Product Management in DoIT User Service to associate your account with the UW-Madison Road Runner License
- Go to the License Center: <https://www.mathworks.com/licensecenter/>
- Follow this tutorial to install and active RoadRunner
 - https://www.mathworks.com/help/roadrunner/ug/install-and-activate-roadrunner.html?s_tid=srchttitle
 - Detailed Guide also available in Google Drive

Open Street Map

Step 1: Click “Export” on the top ⇒ click “Manually”

 STEP 1.

Export

43.07725
-89.40526
43.07125

Manually select a different area

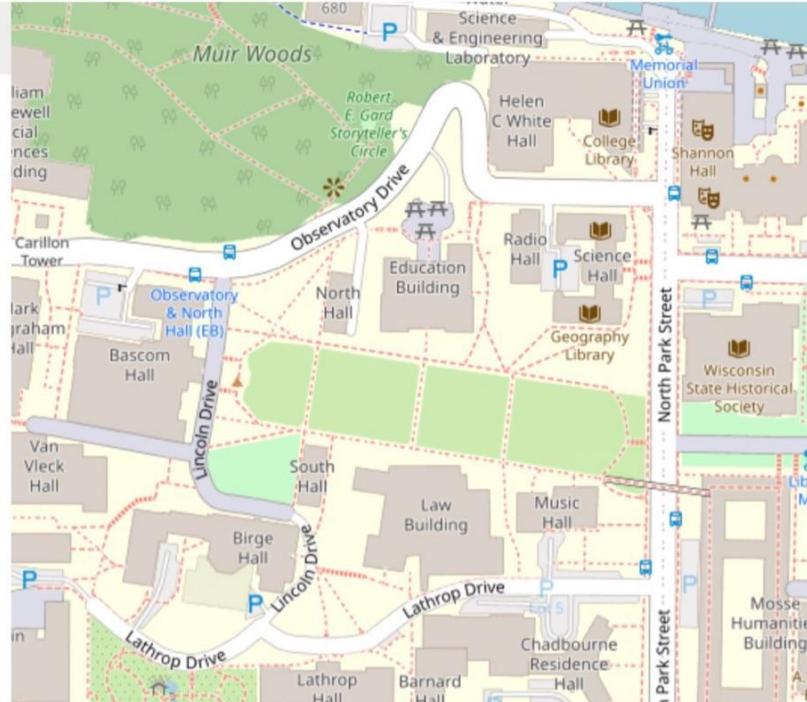
STEP 2

Licence

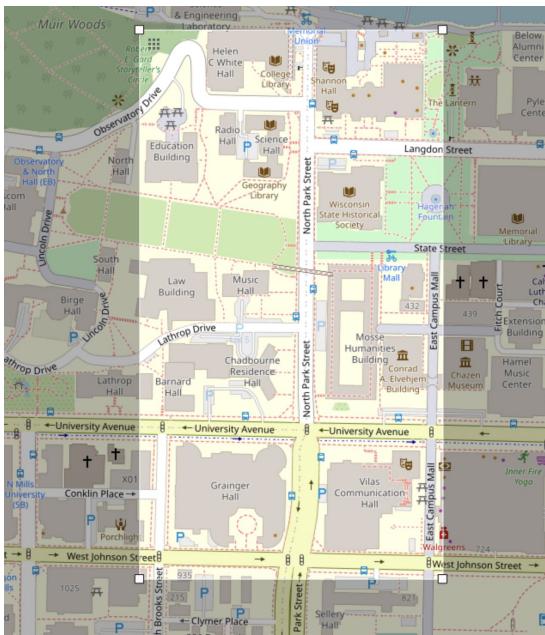
OpenStreetMap data is licensed under the [Open Data Commons Open Database License \(ODbL\)](#).

Export

If the above export fails, please consider using one of the sources listed below:



Step 2: select the region ⇒ export



Licence

OpenStreetMap data is licensed under the [Open Data Commons Open Database License \(ODbL\)](#).

Export

If the above export fails, please consider using one of the sources listed below:

“.osm” ⇒ “.xodr”

- OSM File Format:
 - Map Data from Openstreetmap
- XODR File Format:
 - OpenDRIVE is an open format specification to describe a road network's logic. Its objective is to standardize the logical road description to facilitate the data exchange between different driving simulators.

“.osm” ⇒ “.xodr”

Windows

```
import io

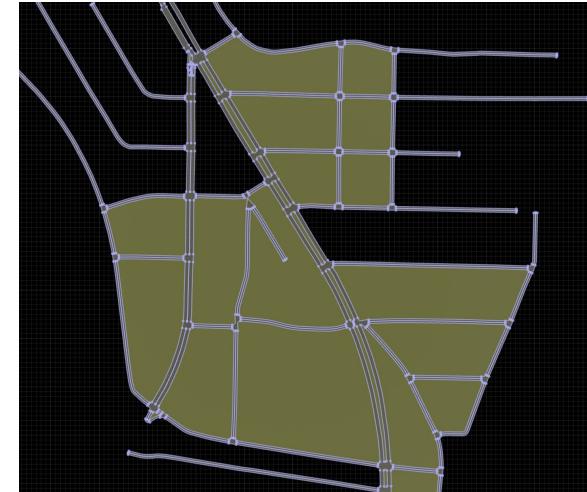
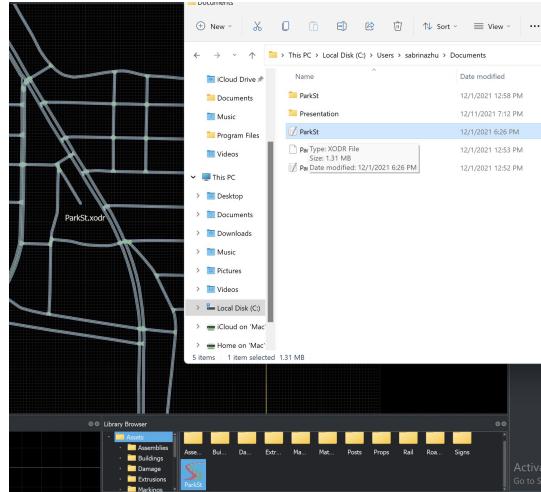
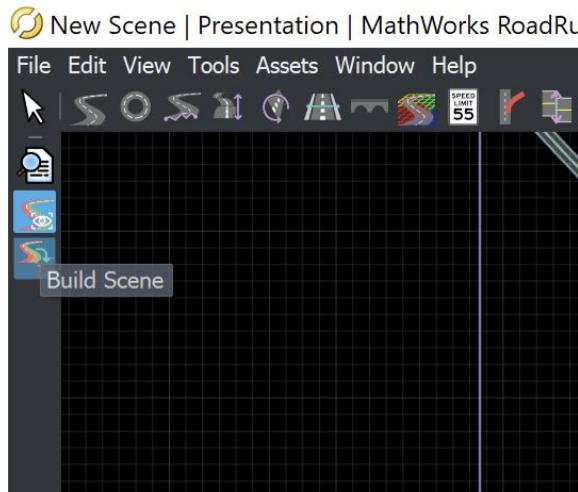
# Read the .osm data
f = io.open("test", mode="r", encoding="utf-8")
osm_data = f.read()
f.close()

# Define the desired settings. In this case, default values.
settings = carla.Osm2OdrSettings()
# Set OSM road types to export to OpenDRIVE
settings.set_osm_way_types(["motorway", "motorway_link", "trunk", "trunk_link", "primary", "primary_link"])
# Convert to .xodr
xodr_data = carla.Osm2Odr.convert(osm_data, settings)

# save opendrive file
f = open("path/to/output/file", 'w')
f.write(xodr_data)
f.close()
```

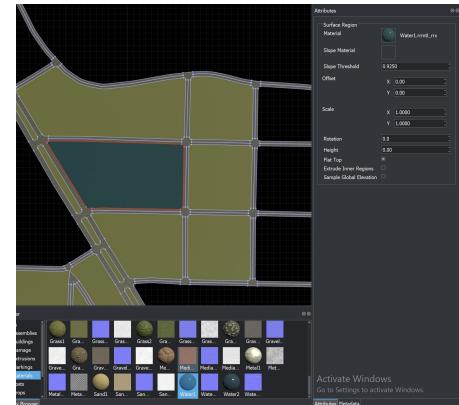
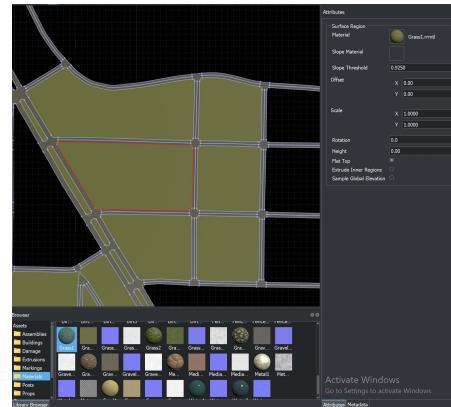
Get the Map

- Drag the map in OpenDrive format to the asset library of RoadRunner
- Drag it into the scene and choose Build Scene
- A street model will then be generated automatically



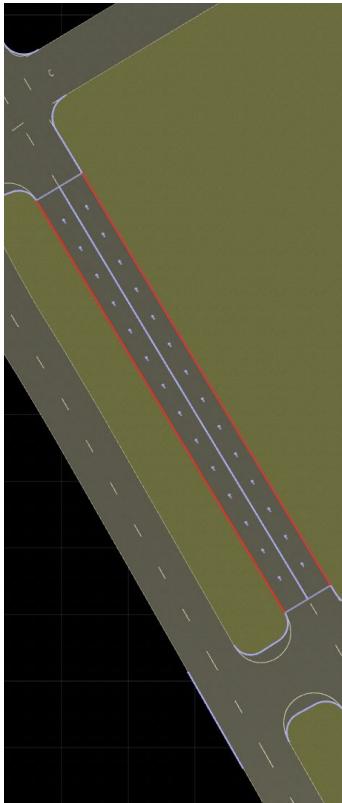
Edit Surface

- Choose “Road Plan Tool” to edit the road or “Surface Tool” to edit other area
- Select the object in the scene whose surface needs to be modified
- All attributes of the selected shape/object will be displayed on the right
- Drag the desired surface material to the top of that shape

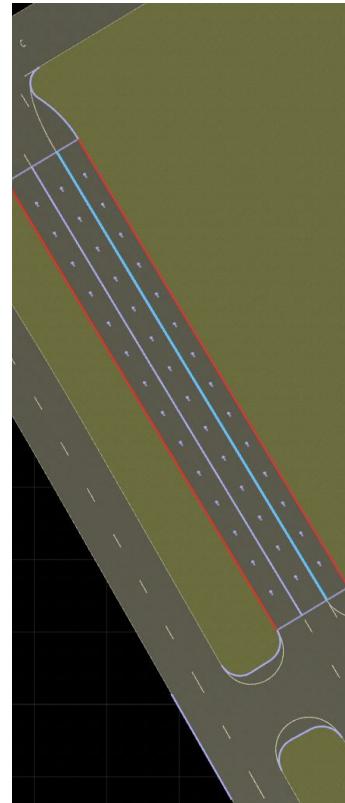


Edit Lanes

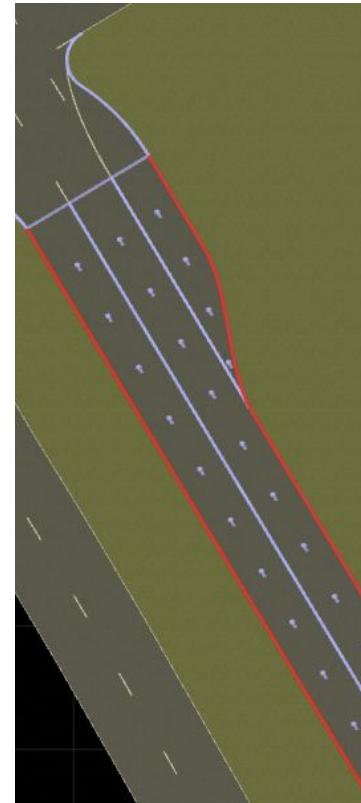
- The “Lane Tool” is used to select a specific lane and edit its attributes
- Add Lane: Use the “Add Lane Tool” to insert new lane of the same length as the current neighboring lanes of that road section.
- Form new lane: Use “Lane Form Tool” to add a lane from a customized starting point on the neighboring lane
 - When there’s a left/right turn lane



Before

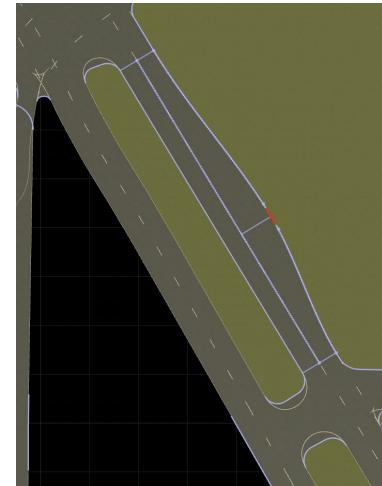
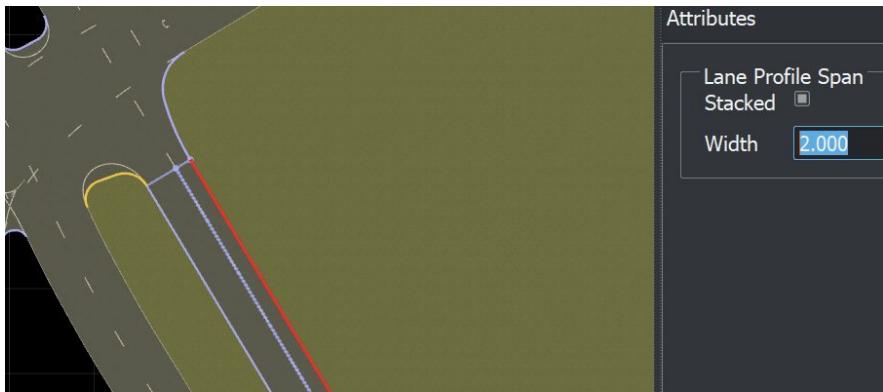
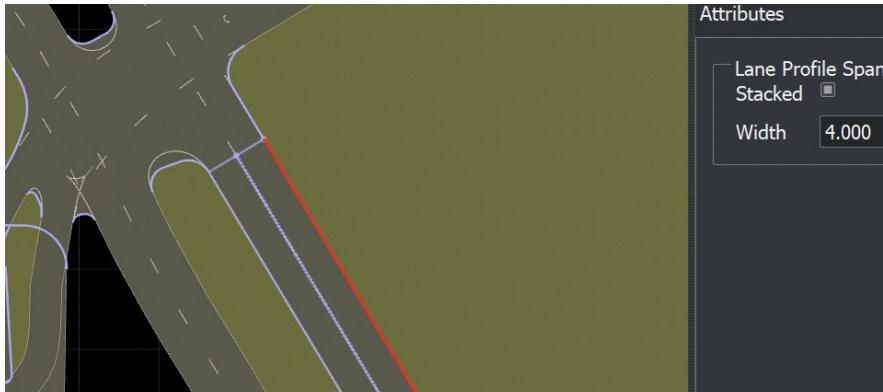


“Add Lane Tool” vs “Lane Form Tool”

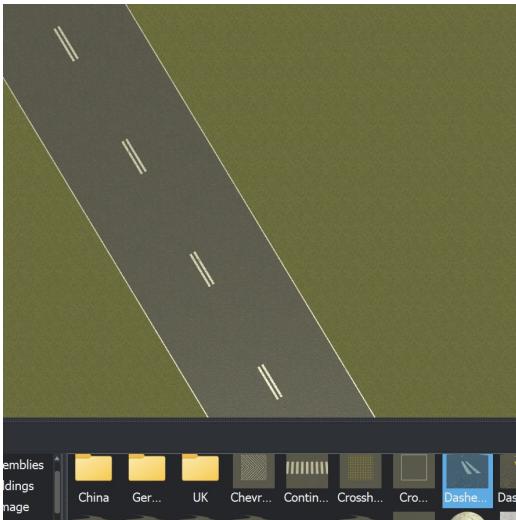
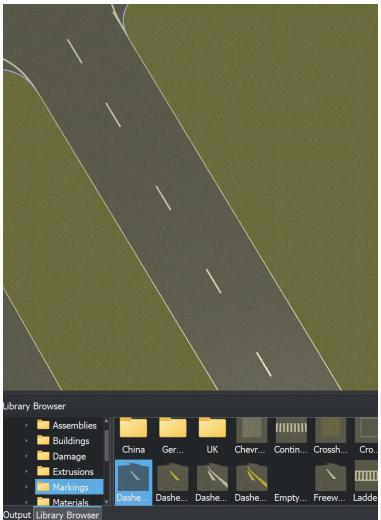


Edit Lanes

- Edit the width with “Lane Width Tool”
- Either by adding markers and drag it to desired width
- Or directly change the value in the attribute panel



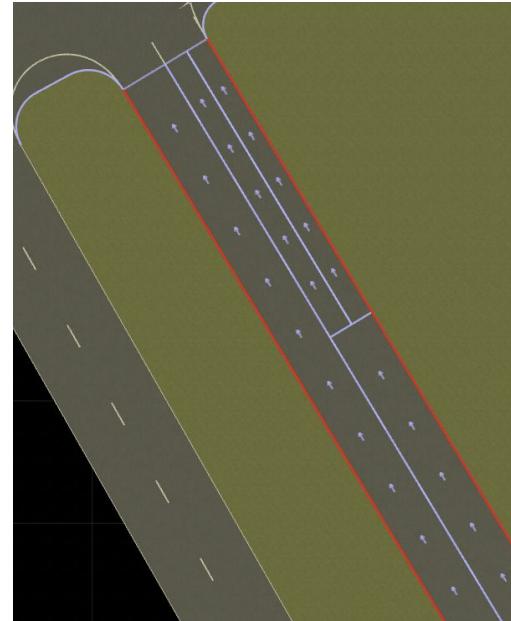
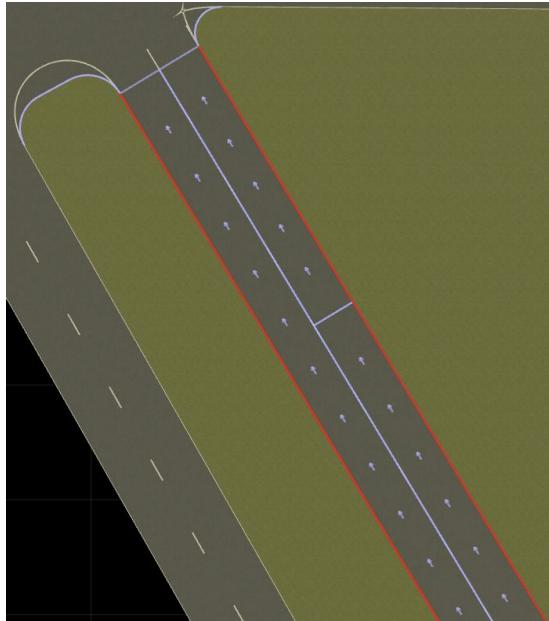
Edit Lanes



- Position of markings can be adjusted by adding markings to drag it around
- Marking type is usually changed by dragging the marking type from the asset library to the scene
- Both can be done with the “Lane Marking Tool”.

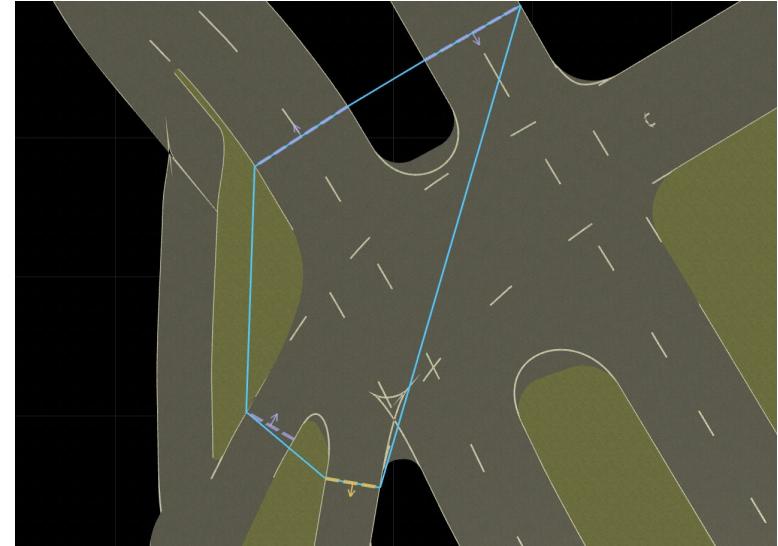
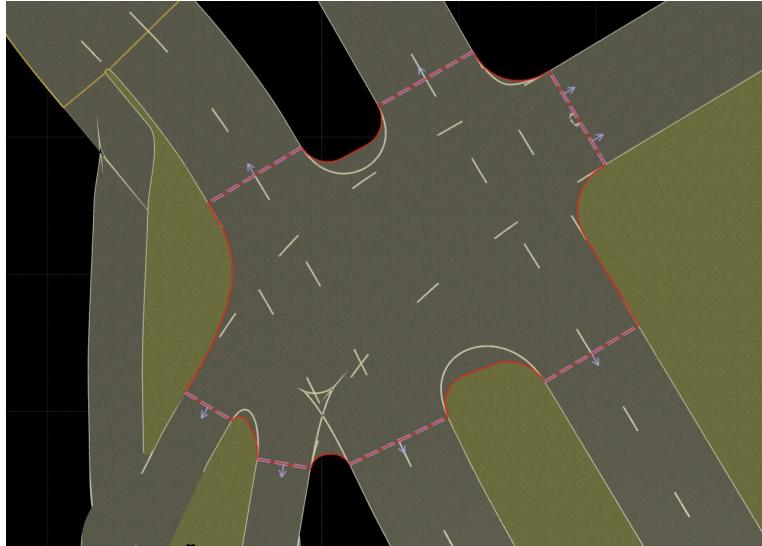
Edit Lanes

- Tools like “Lane Chop Tool” and “Lane Split Tool” can be utilized to separate a single lane into segments and split lanes when needed.



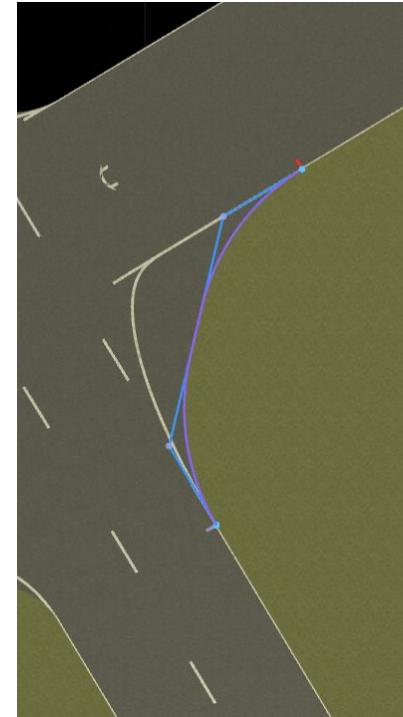
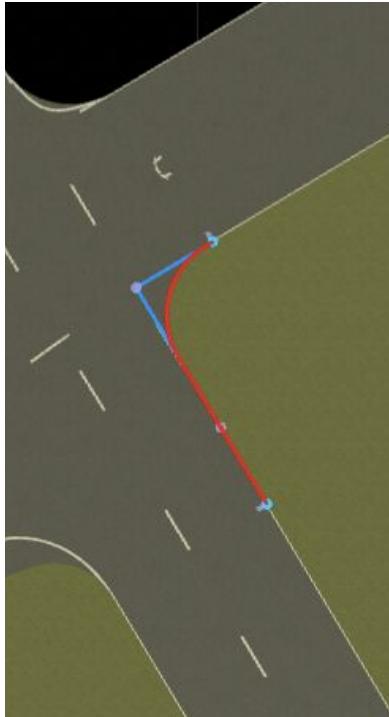
Edit Intersections

- “Custom Junction Tool” is used to create new junctions
- Double click to add a selected road to the junction



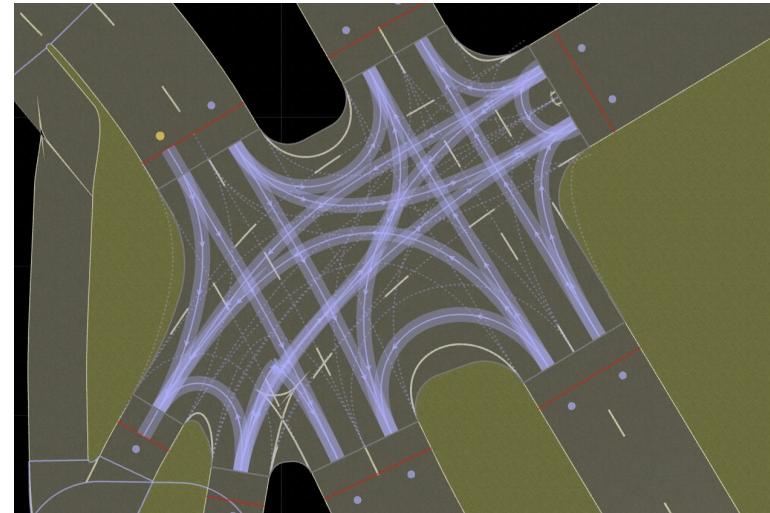
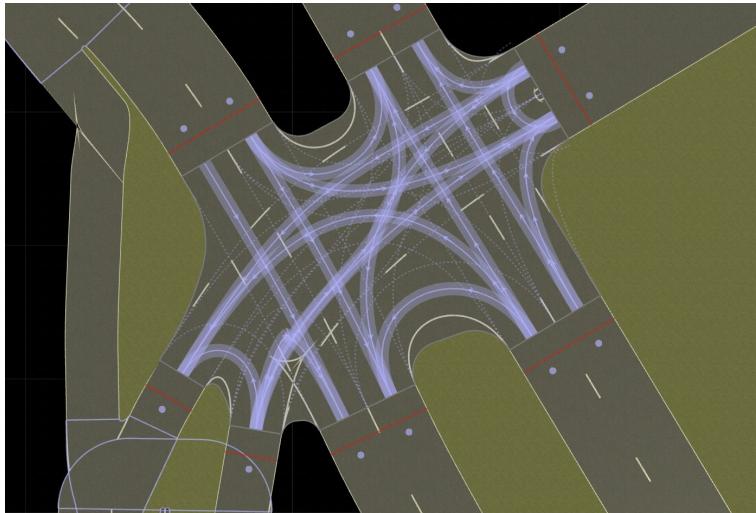
Edit Intersections

- Use the “Corner Tool” to edit the shape of the curve to fit modifications of the lanes
- Click a corner in a intersection to make adjustments
- Drag the marking to change its shape



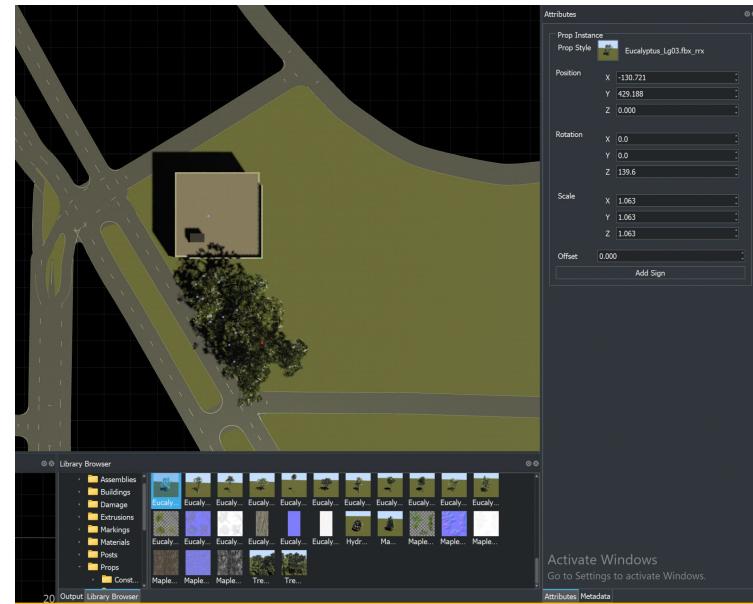
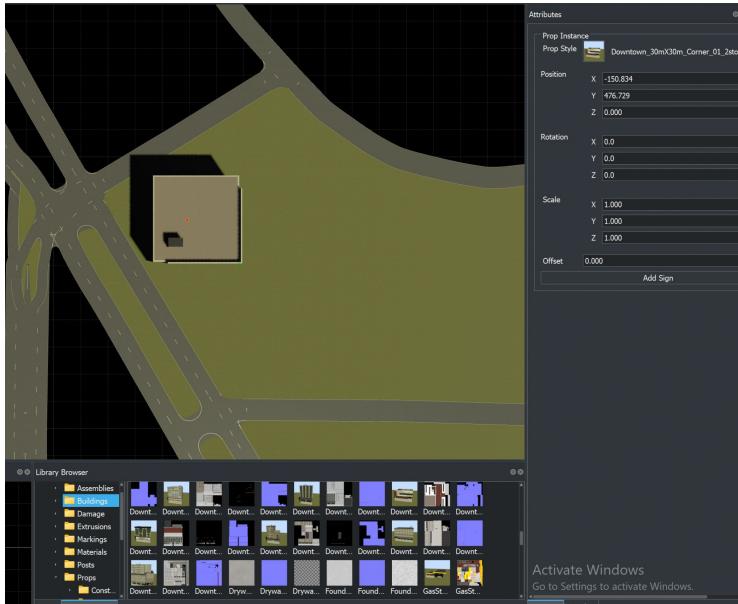
Edit Intersections

- Use the “Maneuver Tool” to edit the maneuver
- Double click on the target location of two lanes to add a maneuver
- Click on a existing maneuver and modify its attributes in the side panel



Add Buildings/Plantations

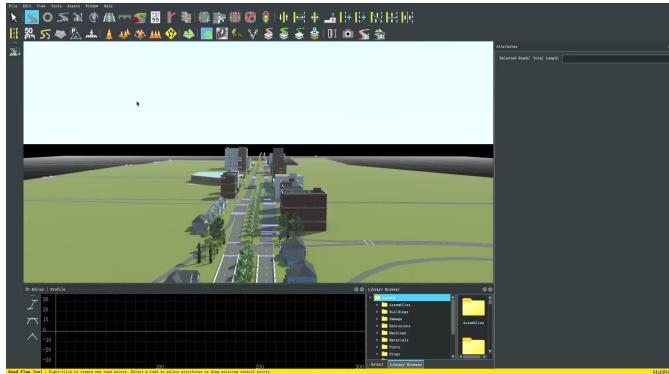
- Do so by dragging the props/building from the asset library to the desired location in the scene



From RoadRunner to CarlaUE4

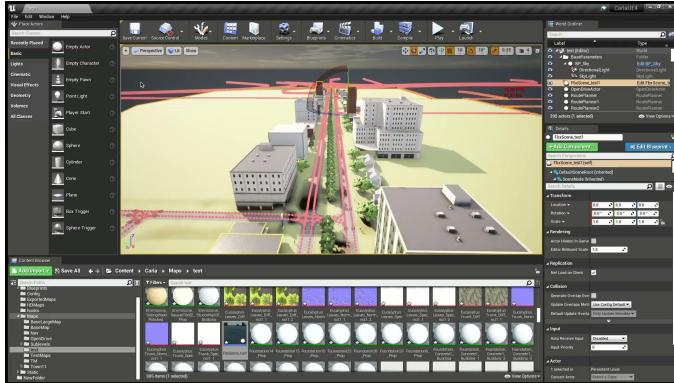
Here what we got in Roadrunner:

Quick Demo:



From RoadRunner to CarlaUE4

Quick Demo:



From RoadRunner to CarlaUE4

Build the environment can always time-consuming and frustrating...

All kinds of errors and bugs need to be fixed:

1. Official tutorial is not up-to-date, and kind of too general.
2. The OS Windows constrain some functions.
3. You even need to edit the source code in carla to solve errors.

```
E:\CARLA>carla>Util>BuildTools>BuildPythonAPI.bat
94 rem
95 rem -----
96 rem -- Check for py
97 rem -----
98 where py 1>nul
99 if %errorlevel% neq 0 goto error_py
100 rem Build for Python 2
101 if %BUILD_FOR_PYTHON2%==true (
102   echo Building Python API for Python 2.
103   python setup.py bdist_egg bdist_wheel
104   if %errorlevel% neq 0 goto error_build_egg
105 )
106 )
107 rem Build for Python 3
108 if %BUILD_FOR_PYTHON3%==true (
109   echo Building Python API for Python 3.
110   python setup.py bdist_egg
111   if %errorlevel% neq 0 goto error_build_egg
112 )
113 )
114 )
115 goto success
116 rem -----
117 rem -- Messages and Errors -----
118 rem -----
119 rem -- Messages and Errors -----
120 rem -----
121 :success
122 echo.
123 if %BUILD_FOR_PYTHON3%==true echo %FILE_N% Car
124 goto good_exit
125
Ln 8, Col 1  Spaces: 4  UTF-8  CRLF  Batch  Late: unsupported
```

Dec 6, 2021 Google Search carla warning C4819 - www.google.com

Dec 6, 2021 Google Search carla make Pythonapi error LNK2019 - www.google.com

From RoadRunner to CarlaUE4

What I will do in the future...

1. Make a detailed tutorial to let team members know how to build CARLA on Windows, with FAQ.
2. Learn how to call CARLA built-in Python APIs
3. Get basic ideas of UE4.

Conclusion

What we have learned

- South Parker Street
- Roadrunner, Carla, Open street Map

What to expect in the future

- Racine? Campus?
- Python APIs
- Traffic Lights and Signs
- Pedestrian Navigation