RWR 4015

Traffic Simulation for Planning Applications

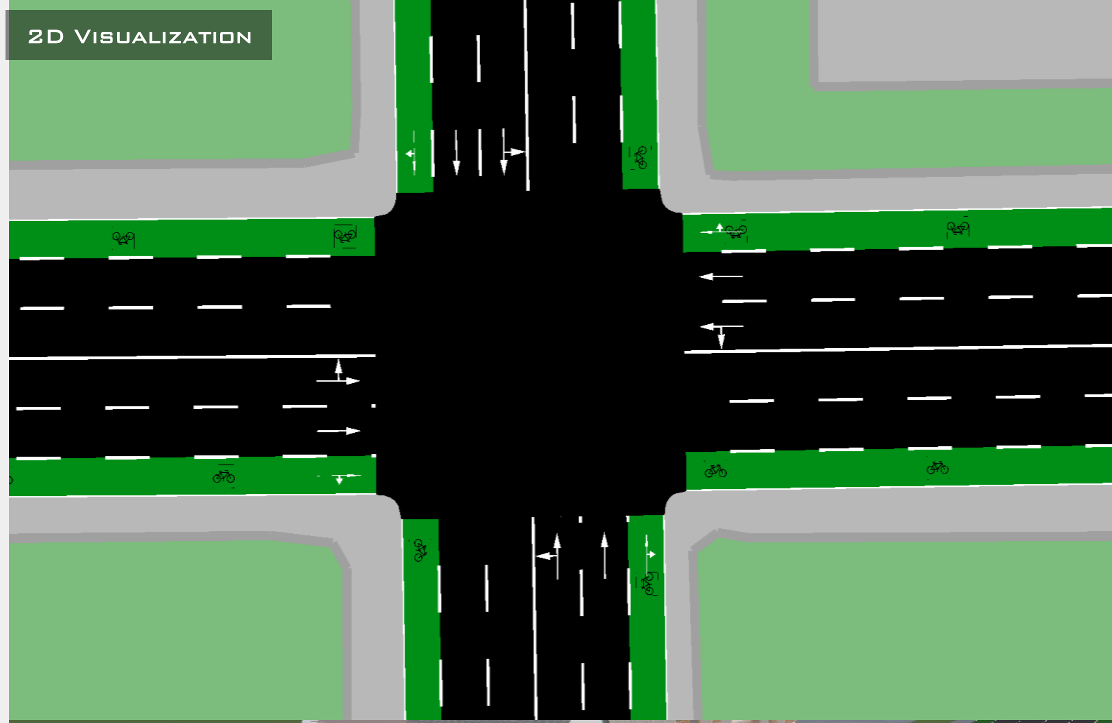
Dr. Ahmad Mohammadi

Week 11 | Lecture:
3D Simulation in Planning II

Fall 2026

RoadwayVR

roadwayvr.github.io/TrafficSimulationforPlanningApplications

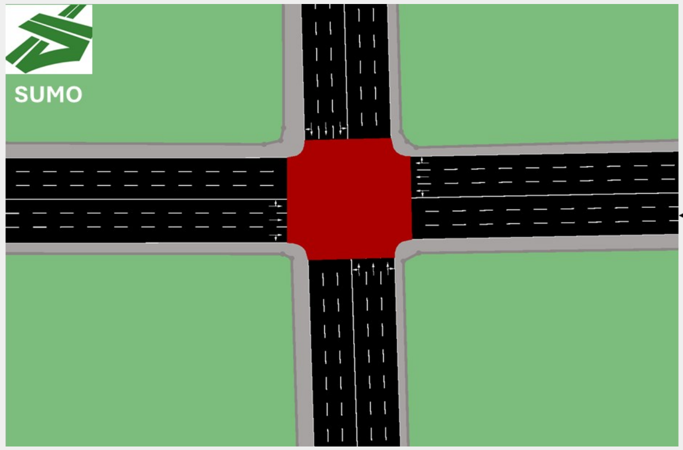


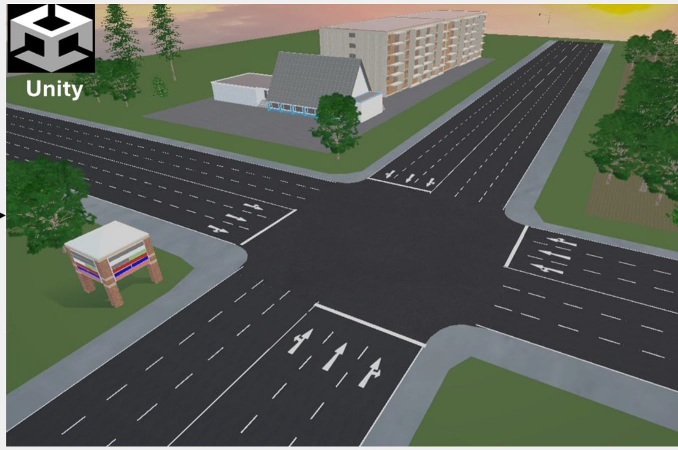
3D Visualization

Scenario 2: (Multi Lane Road with Signalized Intersection)

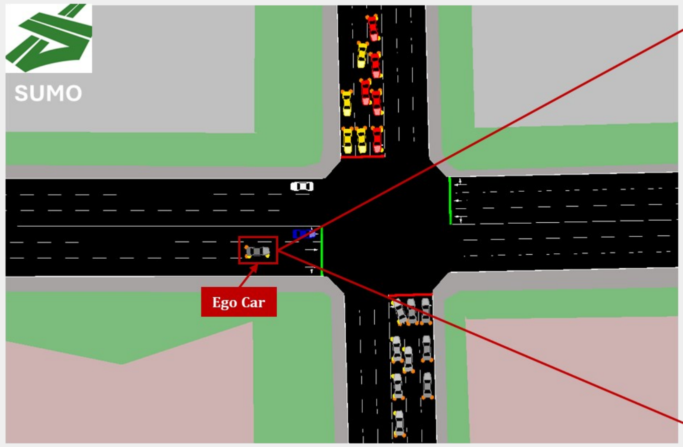
1. Create Road Network
2. Run Sumo2Unity Integration
3. Generate Performance Functions

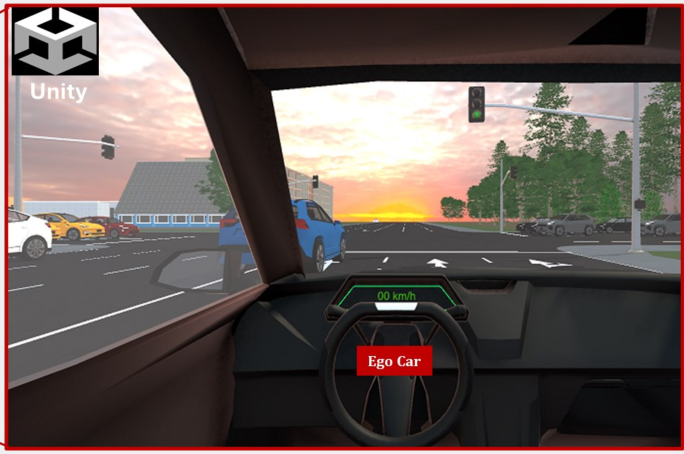
1



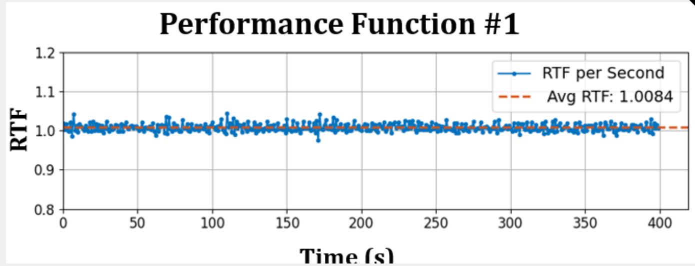


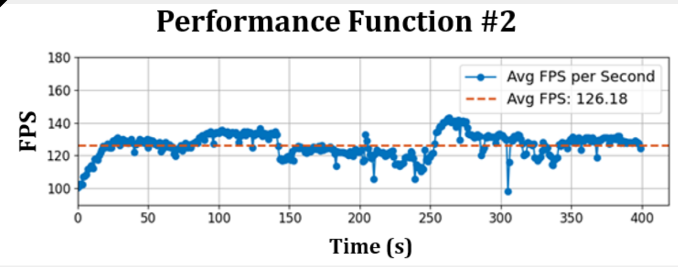
2





3





Step 1: Create Road Network

1.1. SUMO Steps

Note: Create Another Folder as “Scenario2”

A) Adding Lane

B) Adding Terrain

C) Adding Roadside

D) Adding Residential

E) Adding Wood

1.2. Unity Steps

Note: Create a Another Scene as Scenario2

F) Import SUMO Road Network

G) Road Marking As Decals: Stamp an image on a 3D model

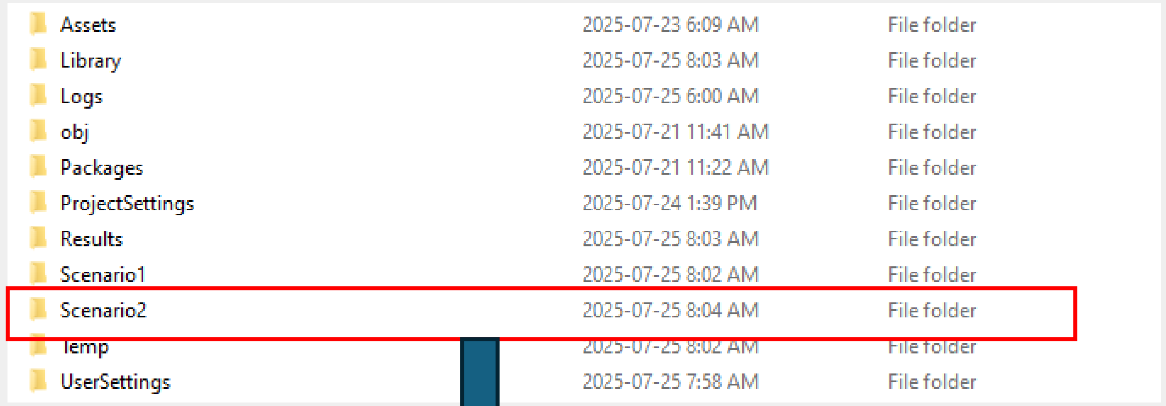
H) Add Stop Signs, and Navigation Arrow

I) Add Trees, Buildings, and Road Signs

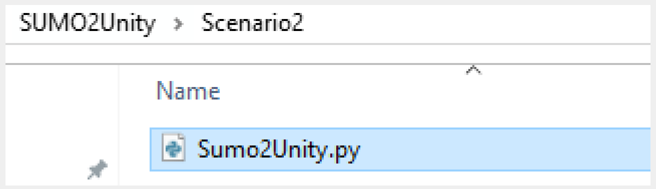
Step 1. Create Road Network

1.1. SUMO Steps:

❑ **Note: Create Another Folder as "Scenario2" and And copy paste Sumo2Unity.py**



Assets	2025-07-23 6:09 AM	File folder
Library	2025-07-25 8:03 AM	File folder
Logs	2025-07-25 6:00 AM	File folder
obj	2025-07-21 11:41 AM	File folder
Packages	2025-07-21 11:22 AM	File folder
ProjectSettings	2025-07-24 1:39 PM	File folder
Results	2025-07-25 8:03 AM	File folder
Scenario1	2025-07-25 8:02 AM	File folder
Scenario2	2025-07-25 8:04 AM	File folder
Temp	2025-07-25 8:02 AM	File folder
UserSettings	2025-07-25 7:58 AM	File folder



SUMO2Unity › Scenario2

Name

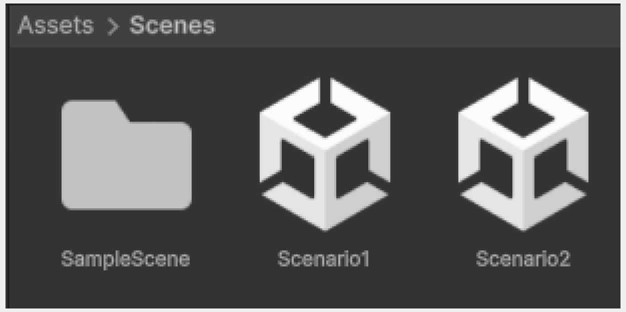
Sumo2Unity.py

1.2. Unity Steps:

❑ **Note: Create Another Scene as "Scenario2"**

❑ Project Window → Scenes → Duplicate Scenario1 (Ctrl + D) → Rename it to Scenario2

❑ Open it and Remove "RoadNetworkRoot" "Decals" and "trees"

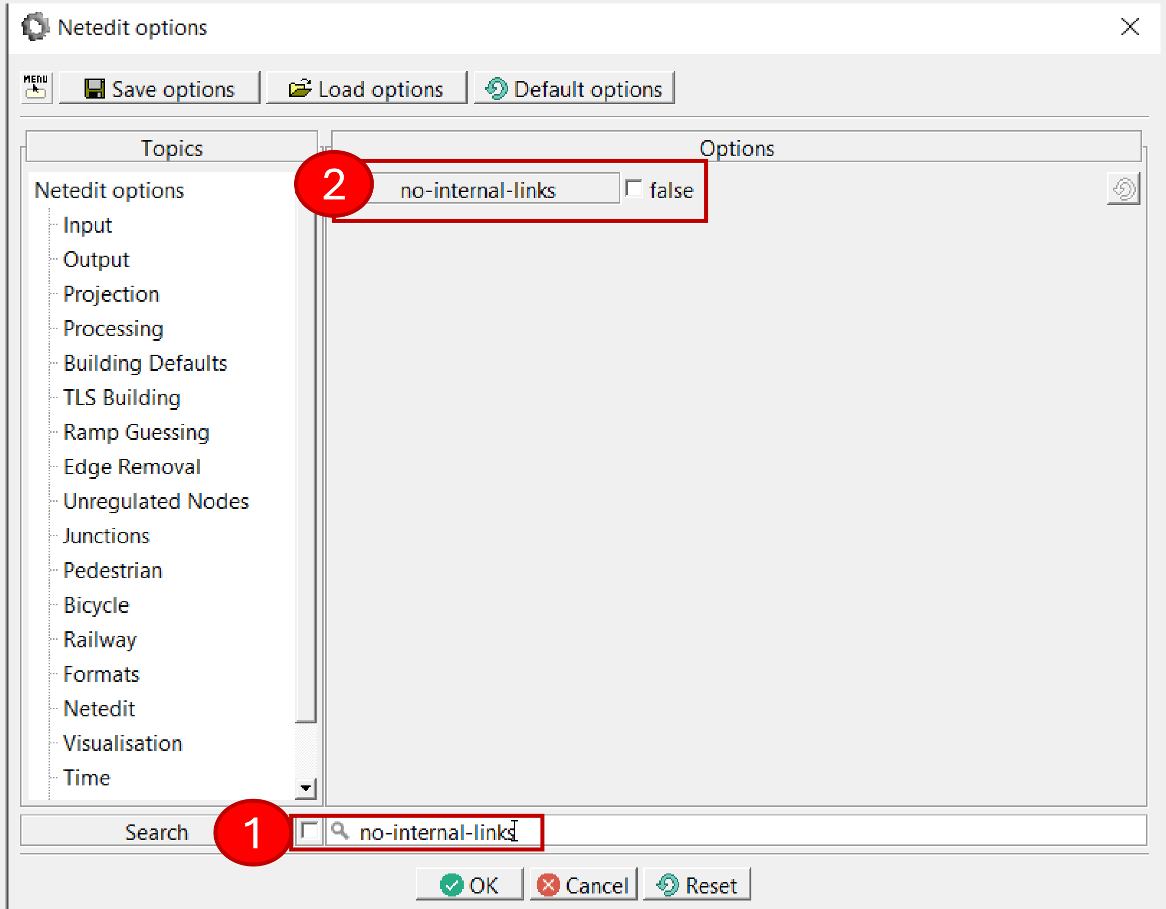


Step 1: Create Road Network

A) Adding Lanes

- ❑ Open netedit → Processing → Option → Search "no-internal-links"

→ Make sure it is like the image



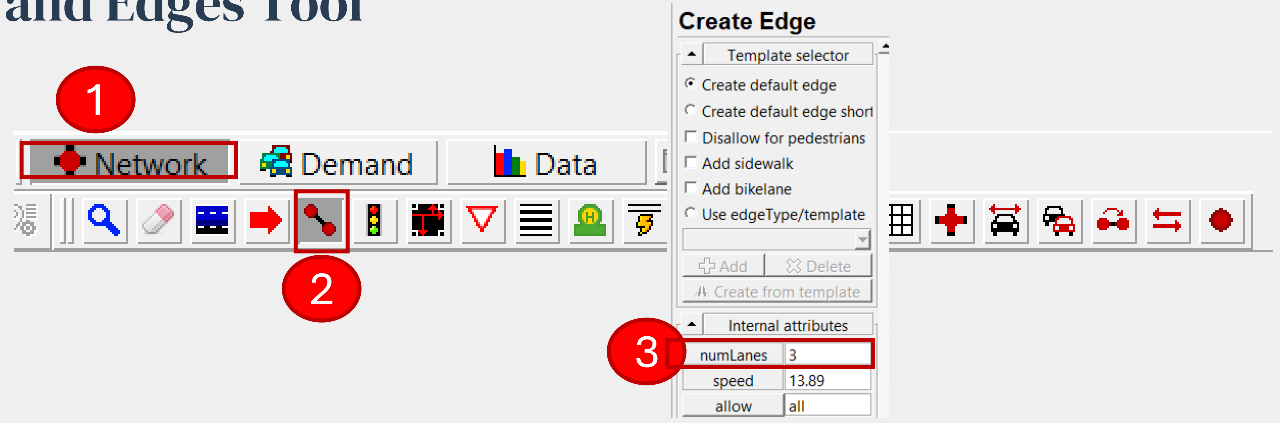
no-internal-links ☐ false

Search: no-internal-links

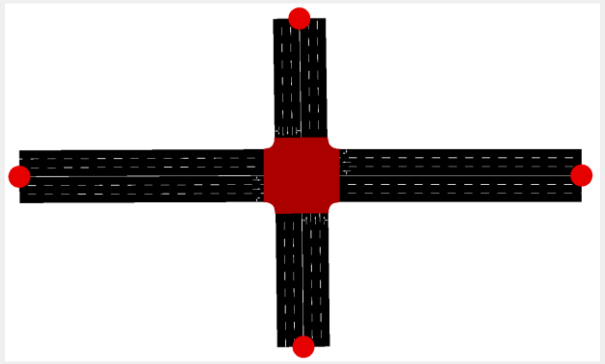
Step 1: Create Road Network

A) Adding Lanes

- ❑ netedit → File → New Network
- ❑ UI → Network → Select “Creating Junction and Edges Tool”
- ❑ Create a simple network like image
- ❑ Add Reverse Direction for All Edges
- ❑ Processing → Compute Junctions



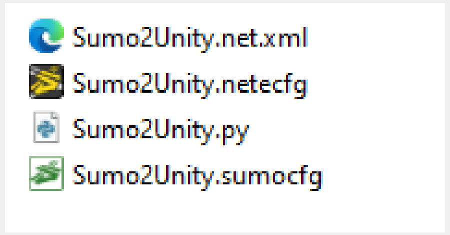
numLanes	3
speed	13.89
allow	all



Step 1: Create Road Network

A) Adding Lanes

- ❑ File → Save Network → Save in Folder "Scenario2" → Name it Sumo2Unity
- ❑ File → Netedit config → Save in Folder "Scenario2" → Name it Sumo2Unity
- ❑ File → Sumo config → Save in Folder "Scenario2" → Name it Sumo2Unity
- ❑ You should have the below files

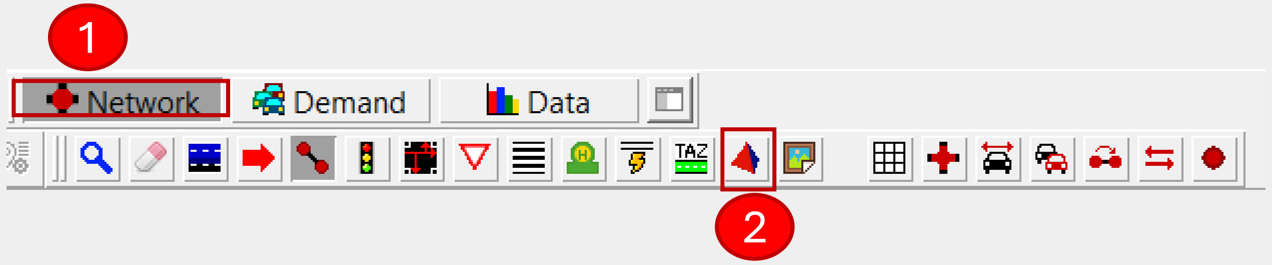


Sumo2Unity.net.xml
Sumo2Unity.netecfg
Sumo2Unity.py
Sumo2Unity.sumocfg

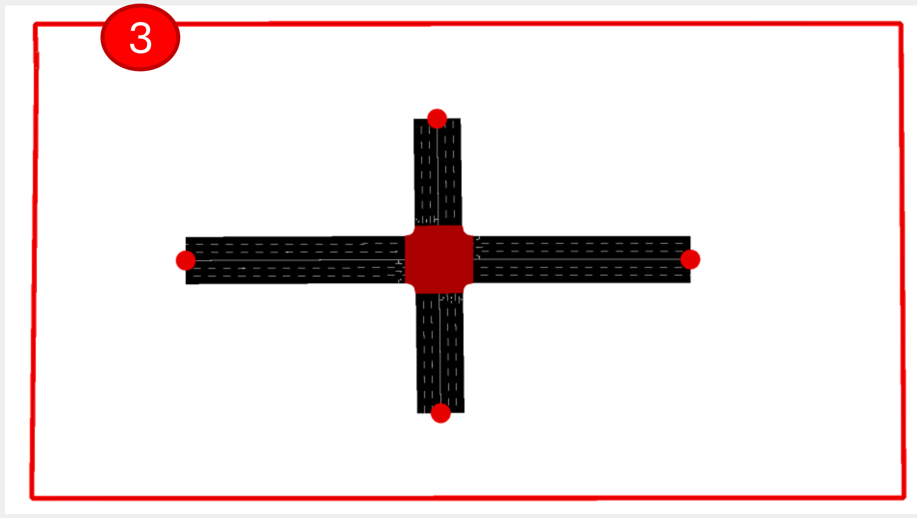
Step 1: Create Road Network

B) Adding Terrain

- ❑ UI → Network → Select "Creating Polygon"



- ❑ Zoom out → in Window "Shapes" (Left Side)→ Type: Terrain → Start Drawing → Create A rectangle that serves as "terrain" → Stop Drawing



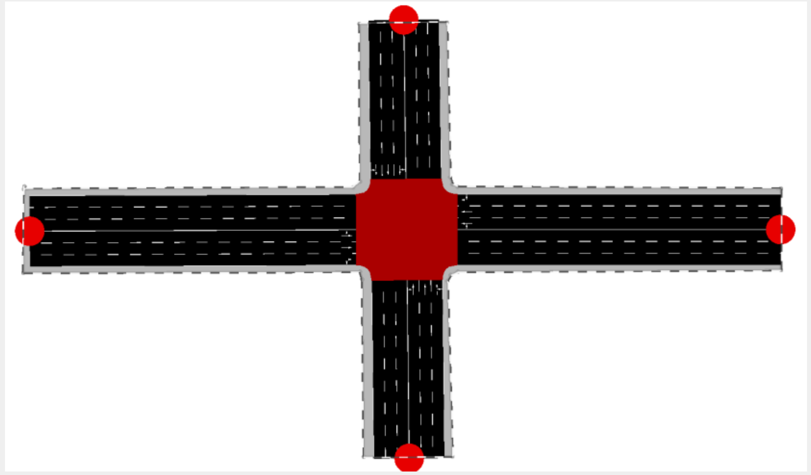
- ❑ File → Additionals and Shapes → Save Additional

→ File Filter: xml files → name as "Sumo2Unity.Poly"

Step 1: Create Road Network

B) Adding Roadside

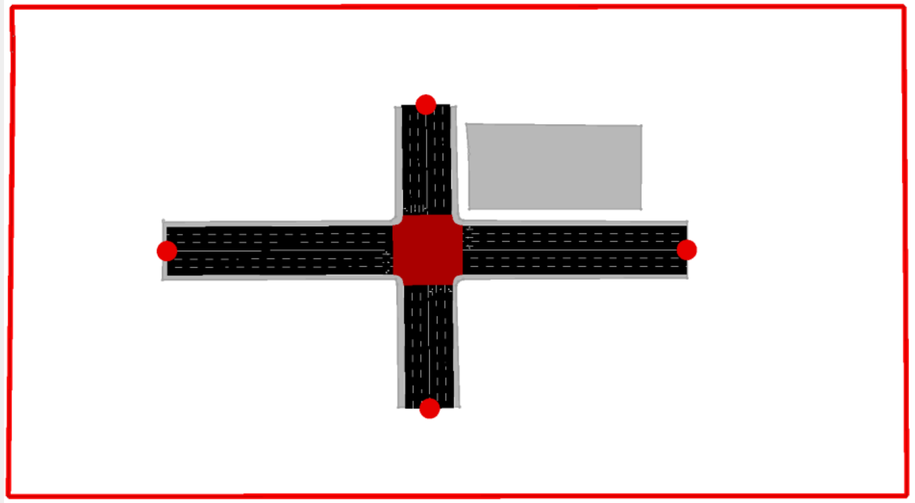
- ❑ UI → Network → Select "Creating Polygon"
- ❑ In Window "Shapes" → Type: Roadside → Start Drawing → Create A "roadside" area that serves as "roadside" → Stop Drawing
- ❑ In Window "Shapes" → Fill: true → color: 122,122,122 (grey)
- ❑ File → Additionals and Shapes → Save Additional



Step 1: Create Road Network

B) Adding Residential

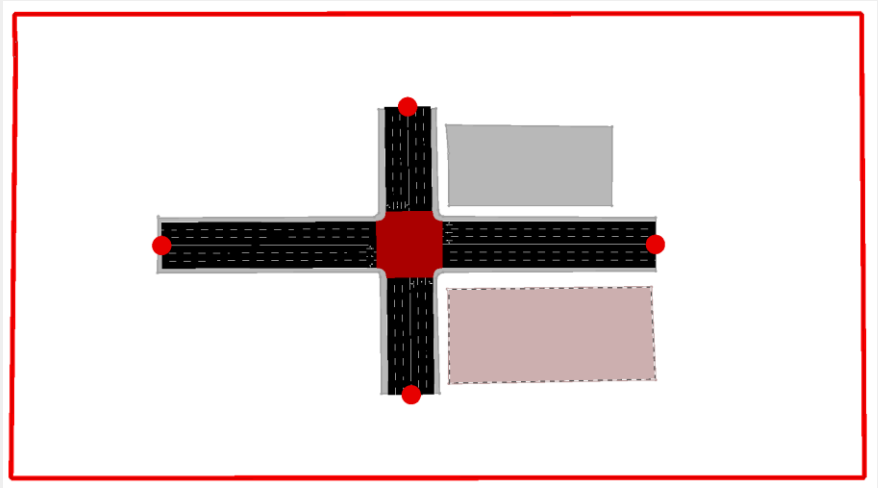
- ❑ UI → Network → Select “Creating Polygon”
- ❑ In Window “Shapes” → Type: Residential → Start Drawing → Create A rectangular that serves as “residential” area → Stop Drawing
- ❑ in Shapes Window → Fill: true → color: 122,122,122 (grey)
- ❑ File → Additionals and Shapes → Save Additional



Step 1: Create Road Network

B) Adding Residential

- ❑ UI → Network → Select “Creating Polygon”
- ❑ In Window “Shapes” → Type: Residential → Start Drawing → Create A rectangular that serves as “wood” area → Stop Drawing
- ❑ In Window “Shapes” → Fill: true → color: 154,110,110
- ❑ File → Additionals and Shapes → Save Additional



Step 1: Create Road Network

1.2. Unity Steps

F) Import SUMO Road Network

G) Road Marking As Decals: Stamp an image on a 3D model

H) Add Stop Signs, and Navigation Arrow

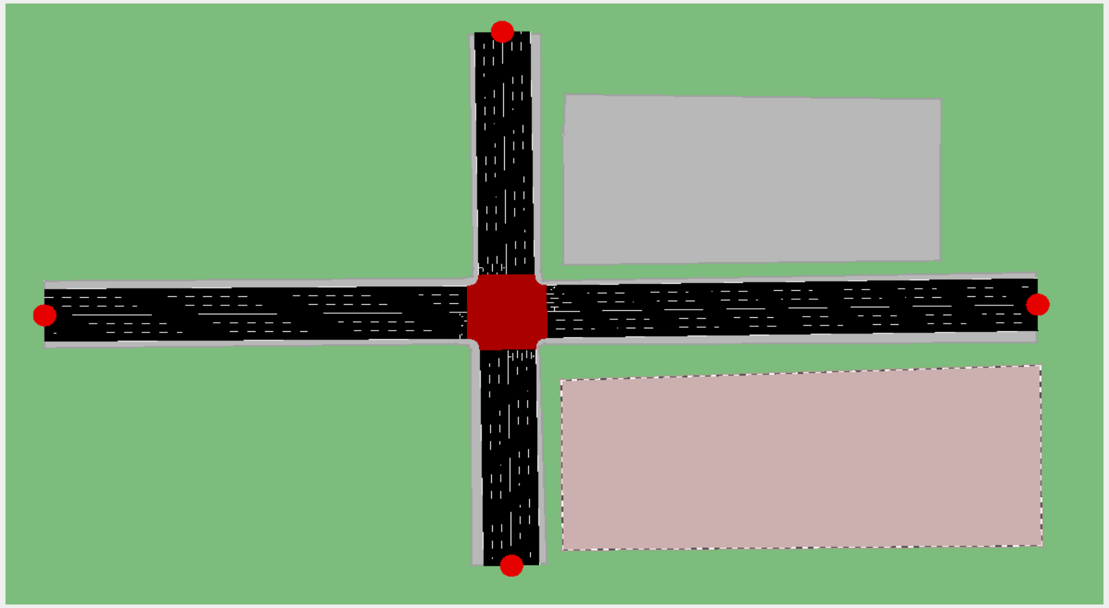
I) Add Trees, Buildings, and Road Signs

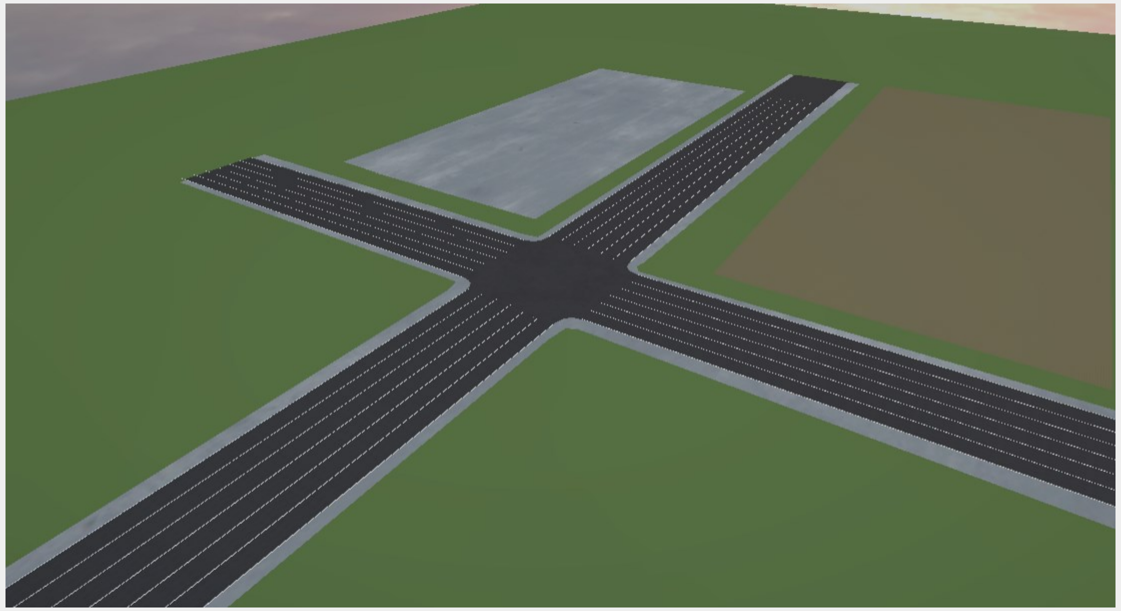
Step 1: Create Road Network

F) Import SUMO Road Network

- ❑ Note: Open Scene “Scenario2”

- ❑ Menu Bar → Sumo2Unity → 1. Create Road Network → Set Sumo Files Folder as Directory\SUMO2Unity\Scenario2 → Start



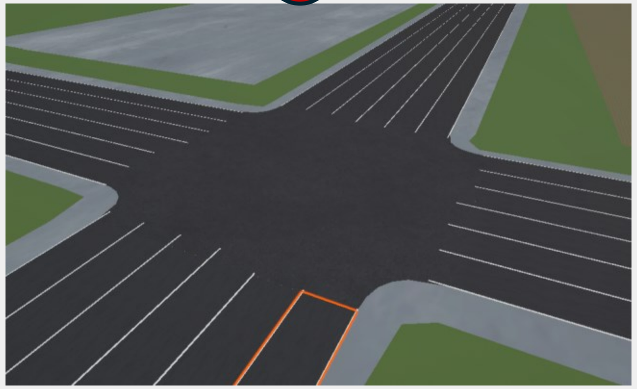


Step 1: Create Road Network

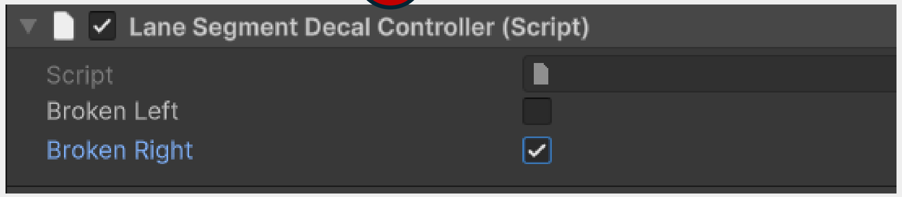
G) Road Marking As Decals: Stamp an image on a 3D model

- ❑ Each Lane has a Left and Right Road Marking Lines
- ❑ To achieve a Broken Line: in Scene Window → Select the Lane in the Image 1
- ❑ in Inspector window, Check the box "Broken Right"

1



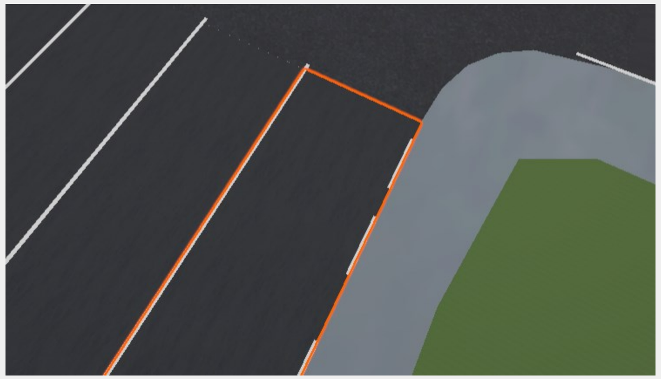
2



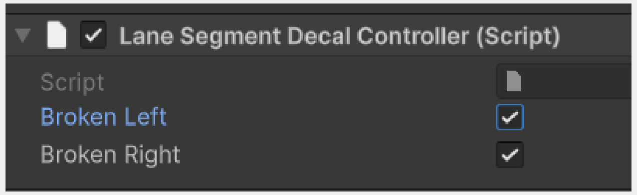
Lane Segment Decal Controller (Script)

Script	
Broken Left	☐
Broken Right	✓

3



- ❑ Check the box "Broken Left" → Seems Nothing Happened Right?



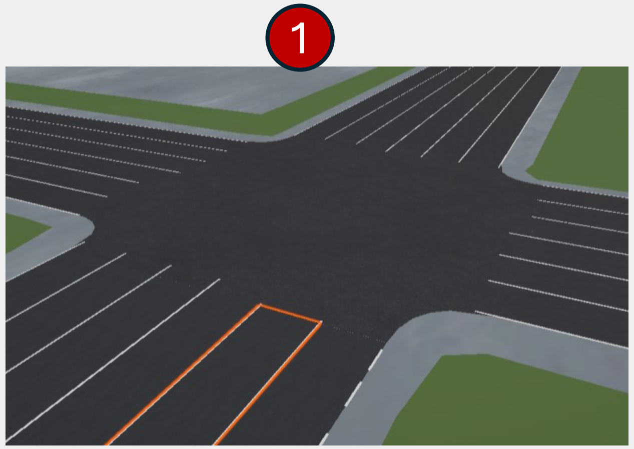
Lane Segment Decal Controller (Script)

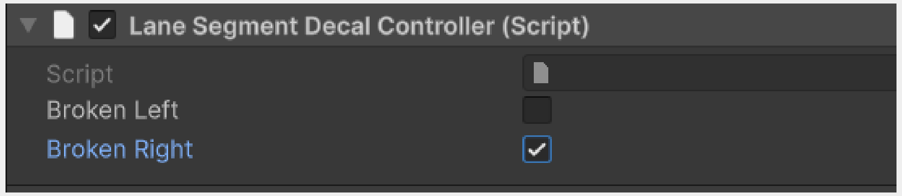
Script	
Broken Left	✓
Broken Right	✓

Step 1: Create Road Network

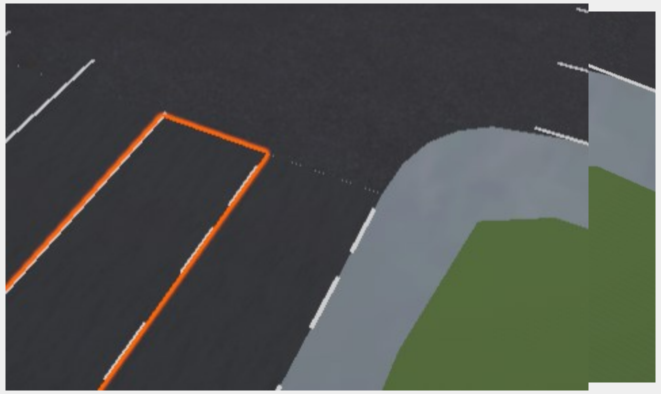
G) Road Marking As Decals: Stamp an image on a 3D model

- ❑ Since each lane has a left and right marking, we need to check the box “Broken Right for middle lane too
- ❑ Do this exercise for the rest of lanes




Lane Segment Decal Controller (Script)

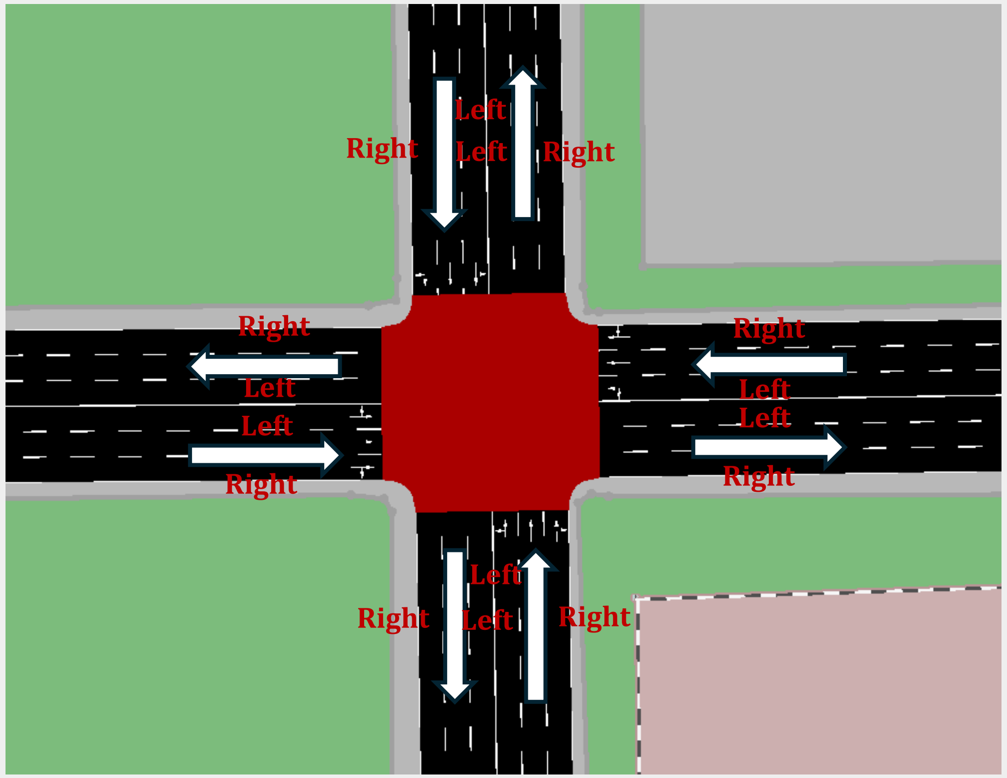
Script	
Broken Left	☐
Broken Right	✓

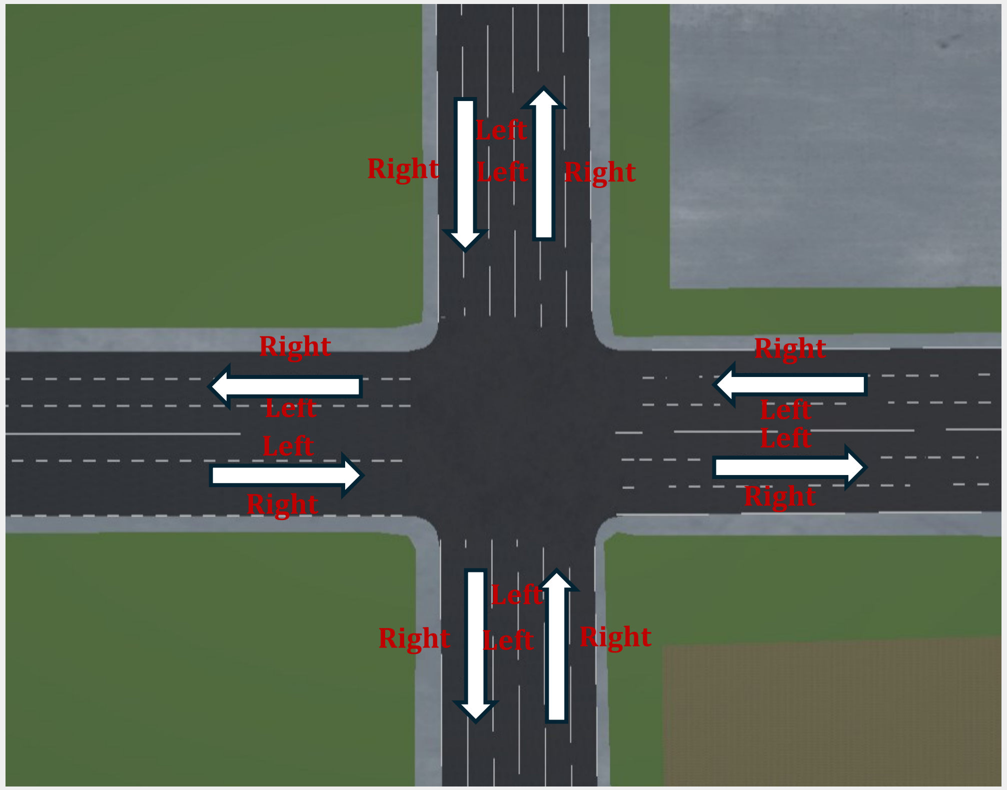


Step 1: Create Road Network

G) Road Marking As Decals: Stamp an image on a 3D model

❑ Hint: Left and Right is the right side and left side of the direction of traveling cars in SUMO

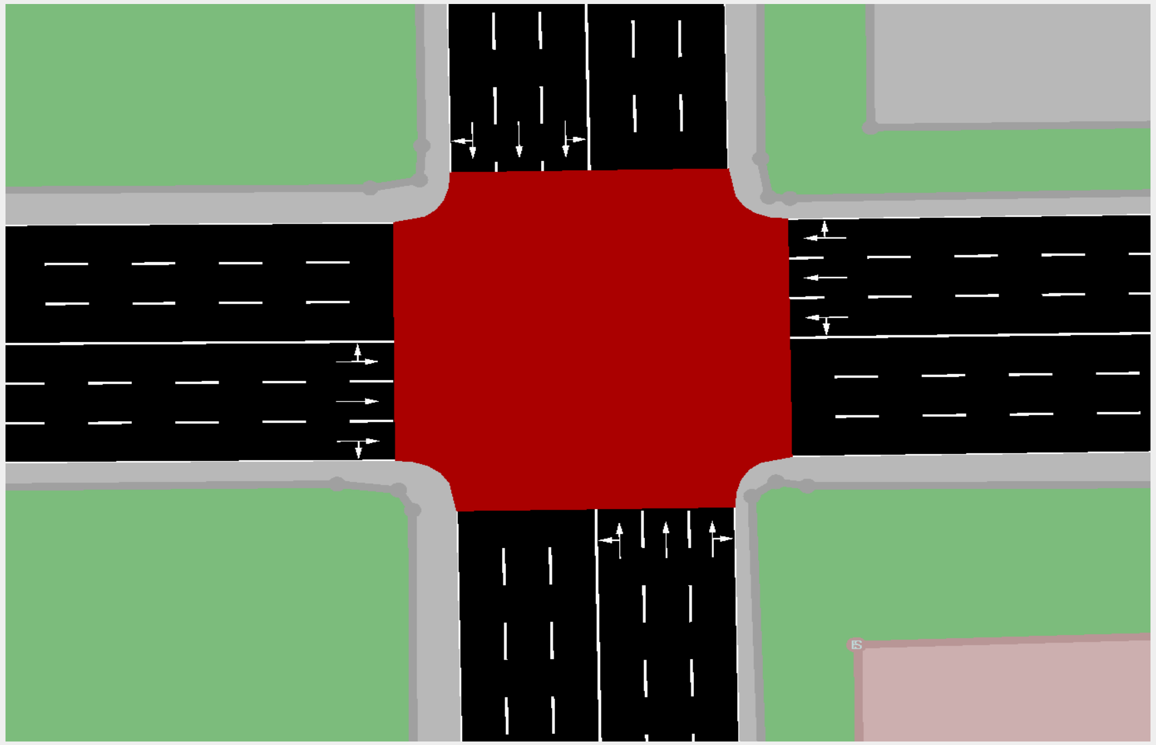


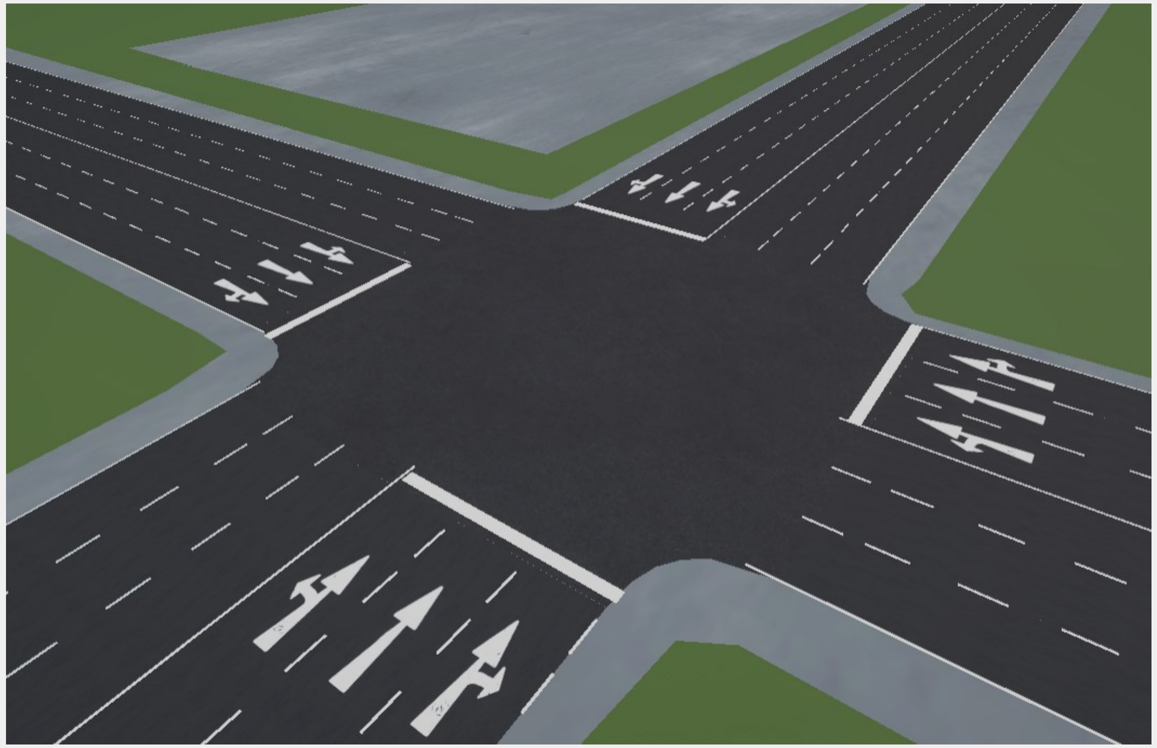


Step 1: Create Road Network

H) Add Stop Sign and Navigation Arrow

- ❑ Hierarchy Window → Rendering → URP Decal Projector

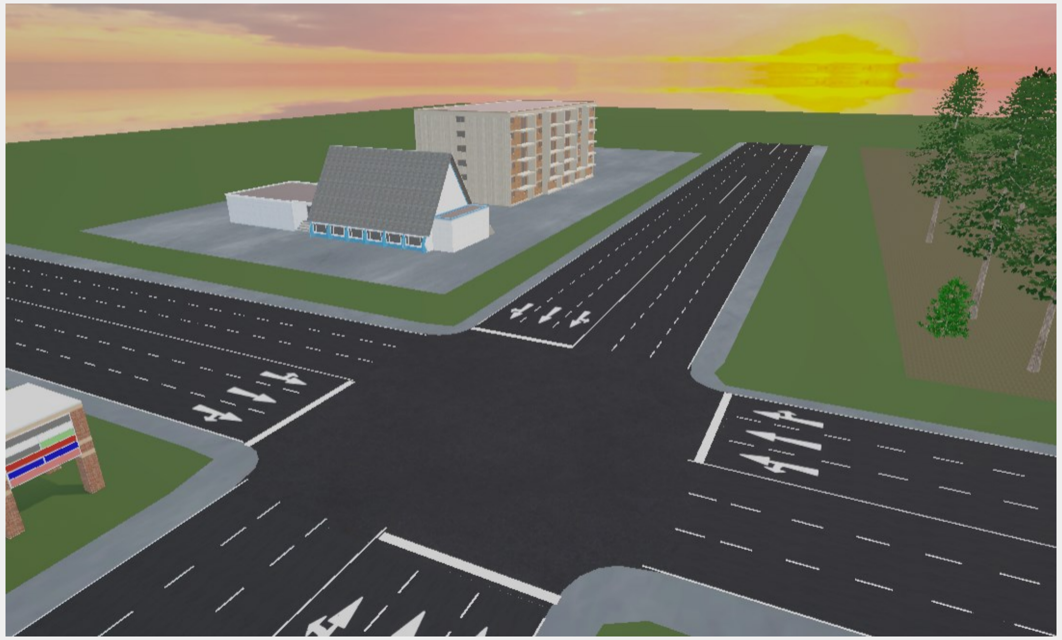




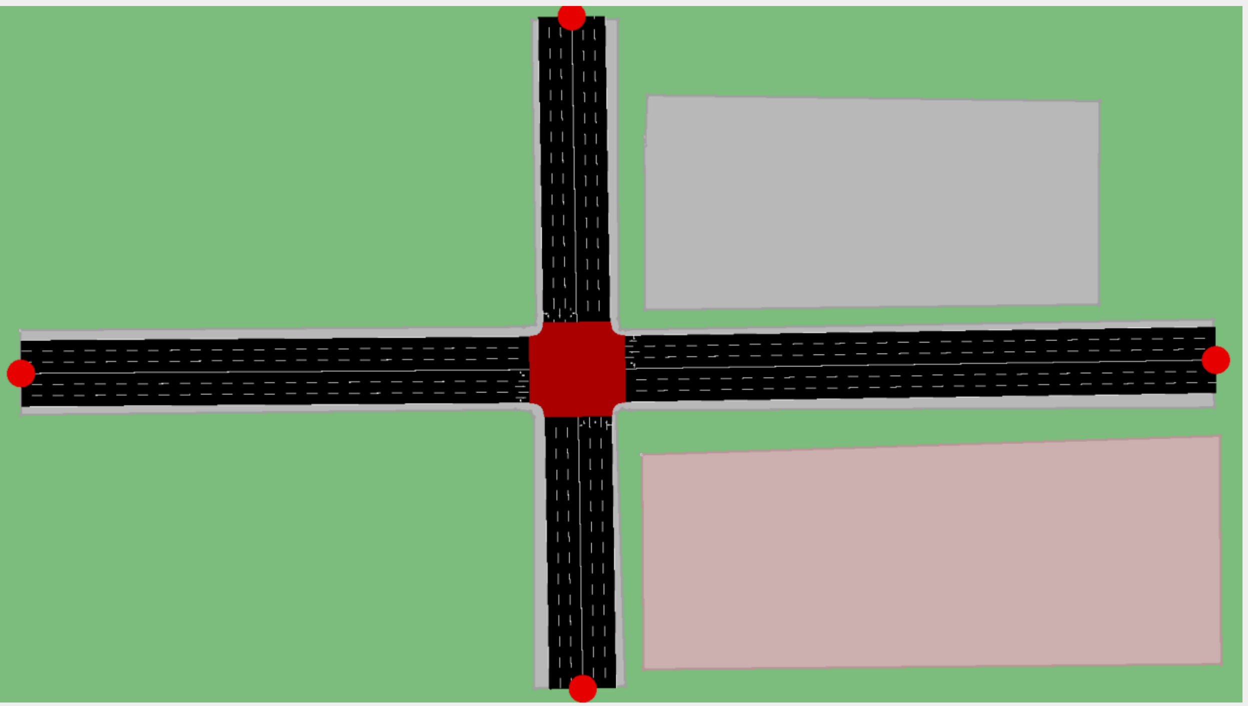
Step 1: Create Road Network

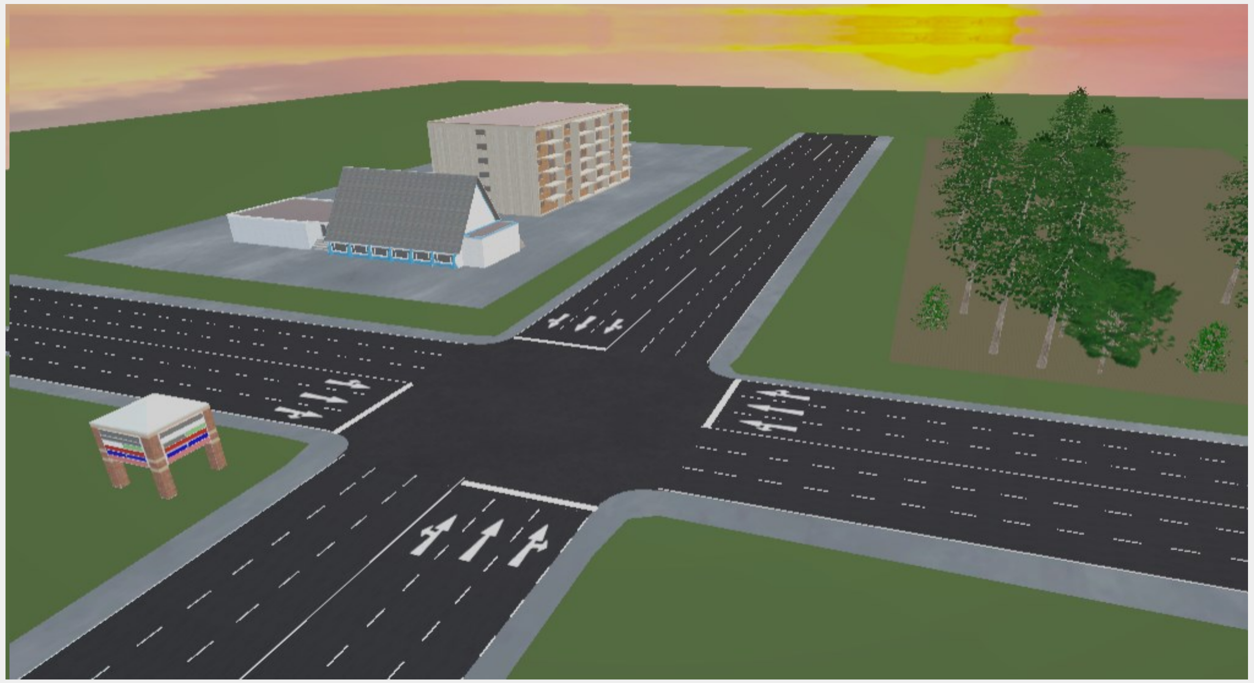
H) Add Trees, Buildings, and Road Signs

- ❑ Project Window → Resources → Trees → Drag and Drop Some Trees in Wood Area
- ❑ Project Window → Resources → Buildings → Drag and Drop Some Buildings in Residential Area
- ❑ Project Window → Resources → Road Signs → Drag and Drop Some Road Signs



Step 1: Create Road Network: Final Output





Step 2: Run Sumo2Unity integration

2.1. SUMO Steps

A) Add Ego Vehicle:

A.1. Create Vehicle Type for EgoCar

A.2. Add Vehicle To Network

B) Add Traffic Volume

B.1. Create Vehicle Types for Traffic Cars

B.2. Add Vehicle To Network

C) Assign Ego Vehicle and Traffic Volume in Unity

D) Prepare and Run Python Code (Sumo2Unity.py)

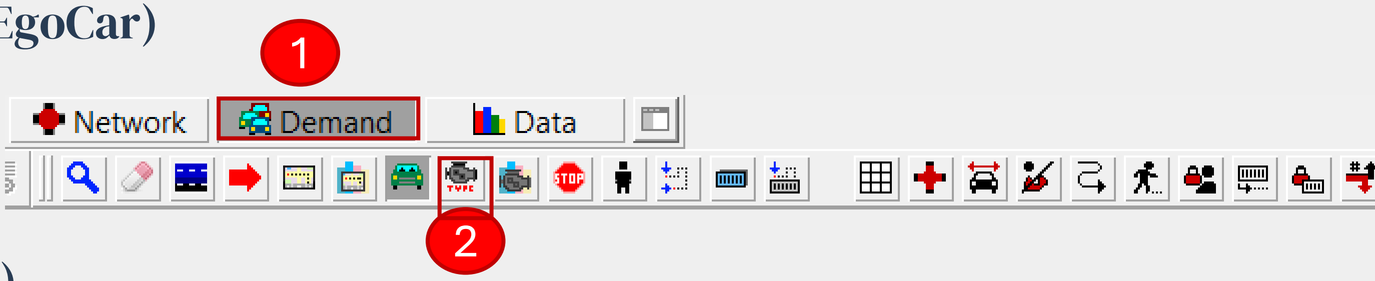
E) Add Traffic Lights in SUMO

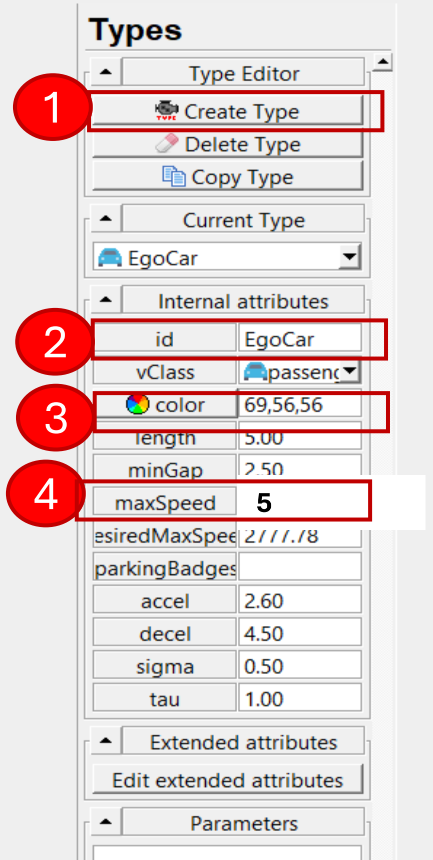
F) Add Traffic Light in Unity

Step 2: Run Sumo2Unity integration

A) Add Ego Vehicle (A.1. Create Vehicle Type for EgoCar)

- ❑ UI → Demand → Select "Creating Vehicles"
- ❑ Create vehicle types EgoCar (Black) (69,56,56)
- ❑ See image
- ❑ File → Demand Element → Save Demand Element → Name it as Sumo2Unity





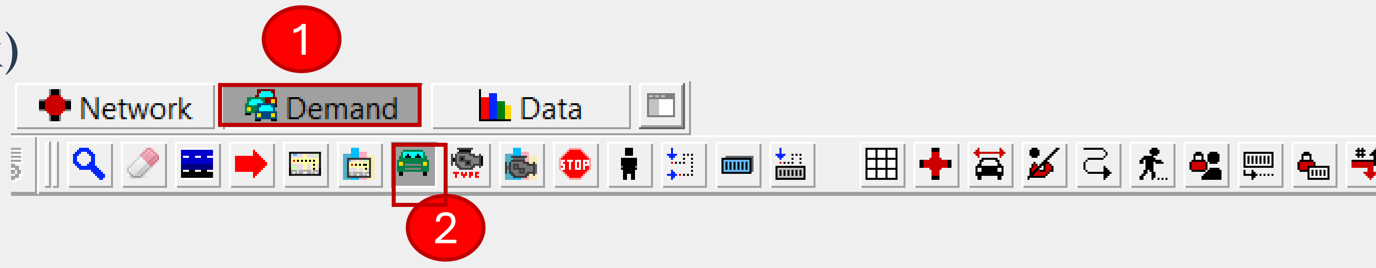
EgoCar

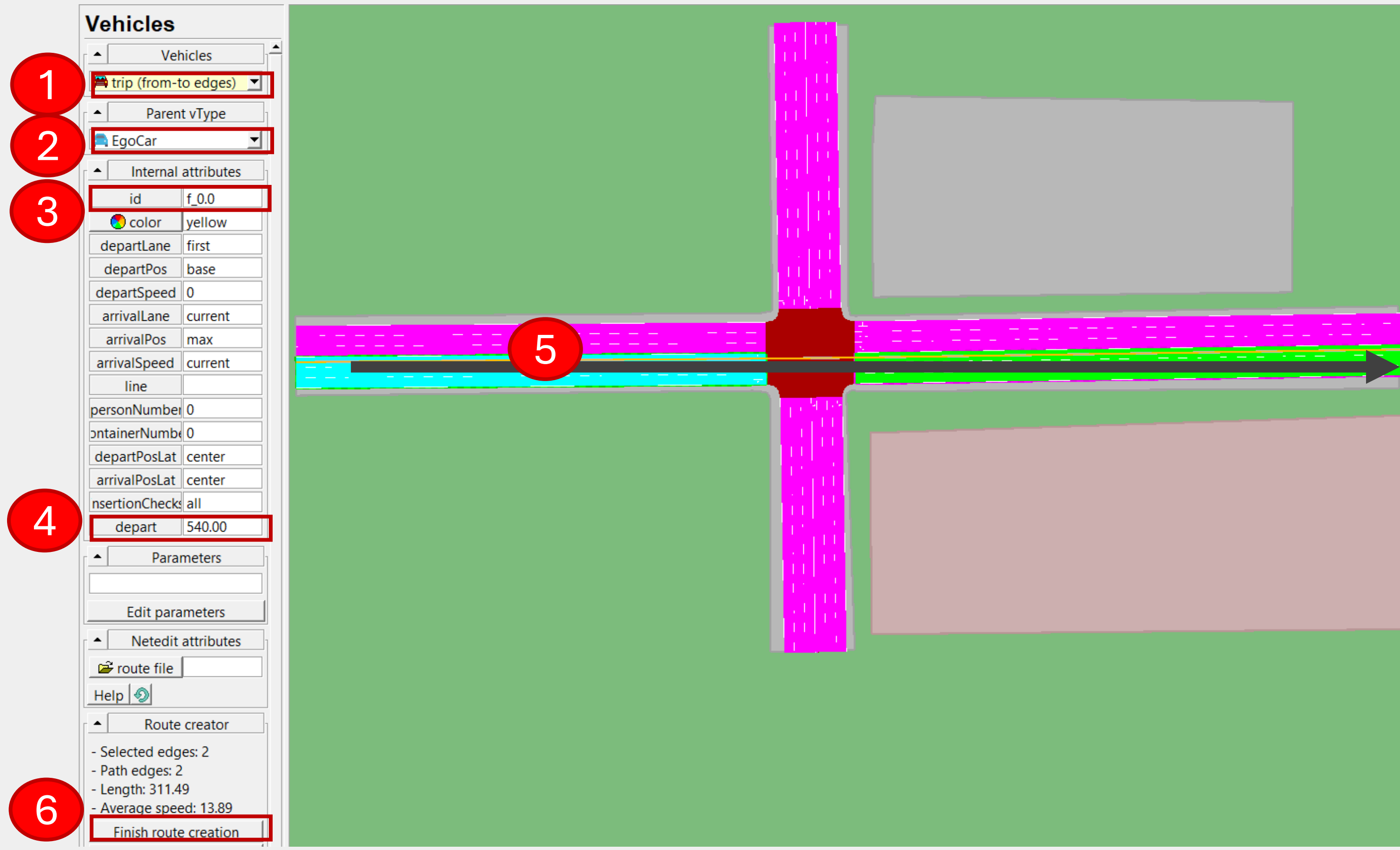
id	EgoCar
vClass	passeng
color	69,56,56
length	5.00
minGap	2.50
maxSpeed	5
esiredMaxSpee	2777.78
parkingBadges	
accel	2.60
decel	4.50
sigma	0.50
tau	1.00

Step 2: Run Sumo2Unity integration

A) Add Ego Vehicle (A.2. Add Vehicle To Network)

- ❑ UI → Demand → Select "Creating Vehicles"
- ❑ Follow steps in Image





Vehicles

Vehicles

trip (from-to edges)

Parent vType

EgoCar

Internal attributes

id	f_0.0
color	yellow
departLane	first
departPos	base
departSpeed	0
arrivalLane	current
arrivalPos	max
arrivalSpeed	current
line	
personNumber	0
ontainerNumbe	0
departPosLat	center
arrivalPosLat	center
nsertionChecks	all
depart	540.00

Parameters

Edit parameters

Netedit attributes

route file

Help

Route creator

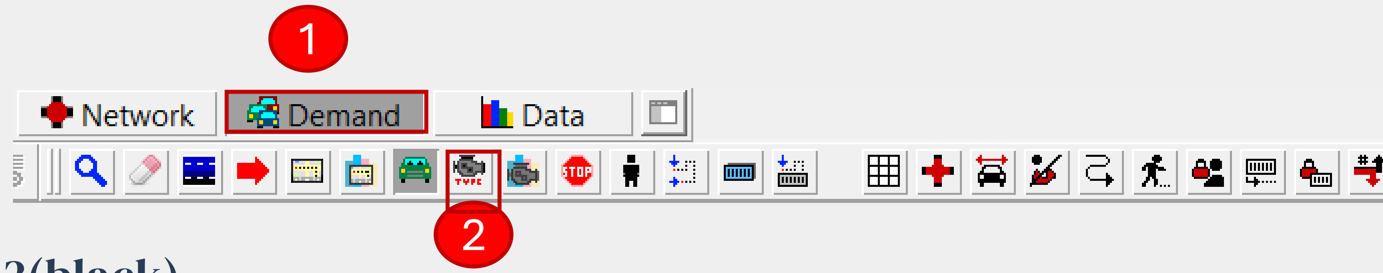
- Selected edges: 2
- Path edges: 2
- Length: 311.49
- Average speed: 13.89

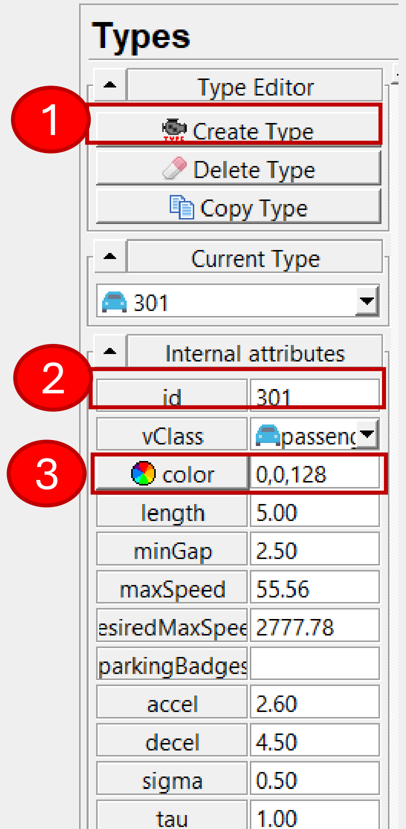
Finish route creation

Step 2: Run Sumo2Unity integration

B) Add Traffic Volume (B.1. Create Vehicle Types for Traffic Cars)

- ❑ UI → Demand → Select "Creating Vehicles"
- ❑ Create vehicle types 301 (blue), 302 (grey), 303(black), 304 (red), 305(gold), 306(white)
- ❑ See 301 (blue as an example)
- ❑ File → Demand Element → Save Demand Element → Name it as Sumo2Unity





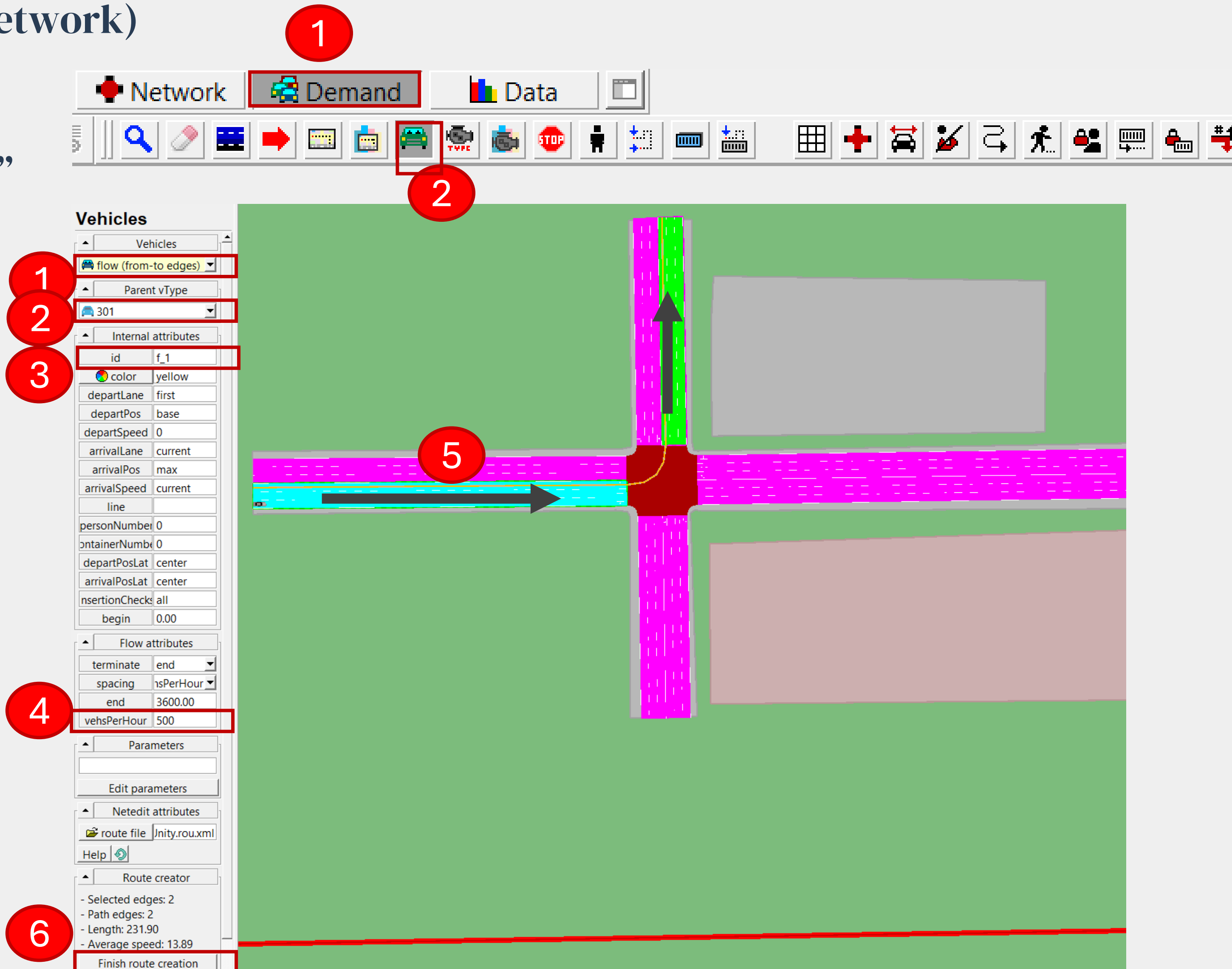
301

id	301
vClass	passeng
color	0,0,128
length	5.00
minGap	2.50
maxSpeed	55.56
esiredMaxSpee	2777.78
parkingBadges	
accel	2.60
decel	4.50
sigma	0.50
tau	1.00

Step 2: Run Sumo2Unity integration

B) Add Traffic Volume (B.2. Add Vehicle To Network)

- ❑ UI → Demand → Select “Creating Vehicles”
- ❑ Follow Steps



Vehicles

flow (from-to edges)

Parent vType

301

Internal attributes

id	f_1
color	yellow
departLane	first
departPos	base
departSpeed	0
arrivalLane	current
arrivalPos	max
arrivalSpeed	current
line	
personNumber	0
ɔntainerNumbe	0
departPosLat	center
arrivalPosLat	center
nsertionChecks	all
begin	0.00

Flow attributes

terminate	end
spacing	ıPerHour
end	3600.00
vehsPerHour	500

Parameters

Edit parameters

Netedit attributes

route file	Jnity.rou.xml

Help

Route creator

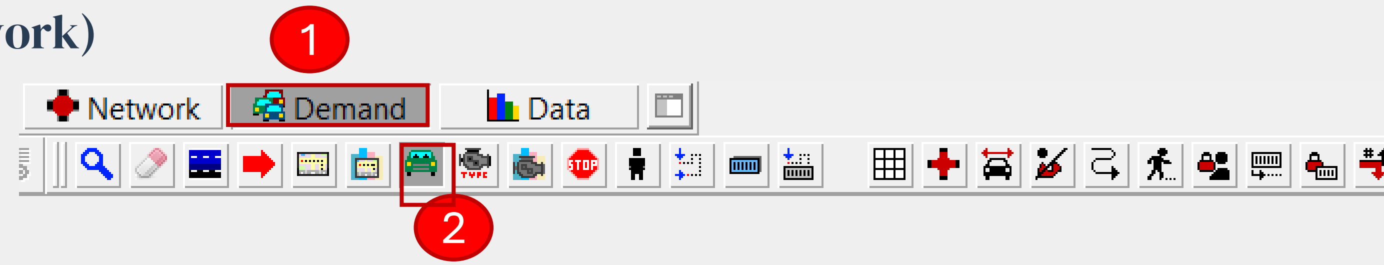
- Selected edges: 2
- Path edges: 2
- Length: 231.90
- Average speed: 13.89

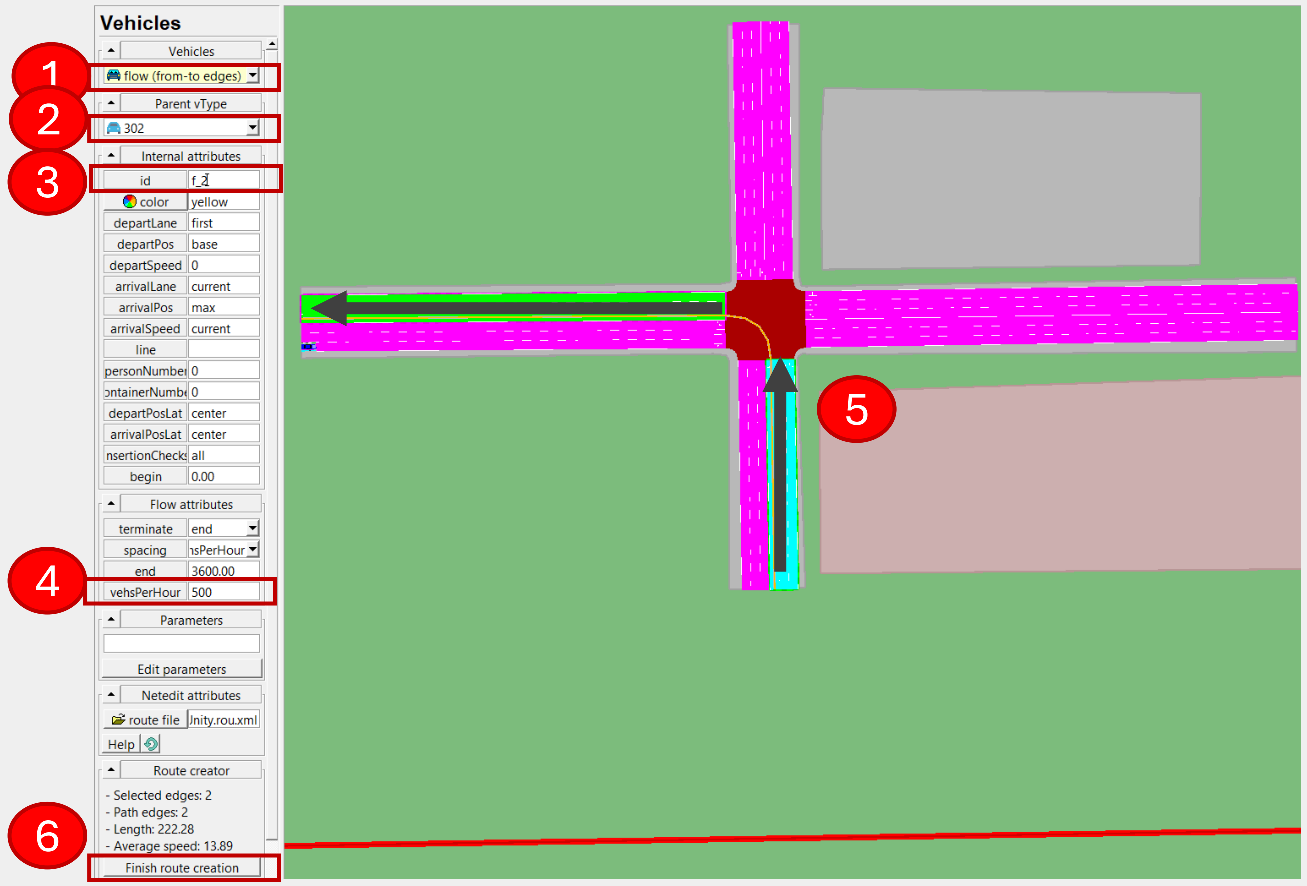
Finish route creation

Step 2: Run Sumo2Unity integration

B) Add Traffic Volume (B.2. Add Vehicle To Network)

- ❑ Do this for 302





Vehicles

Vehicles: flow (from-to edges)

Parent vType: 302

Internal attributes

id	f_2
color	yellow
departLane	first
departPos	base
departSpeed	0
arrivalLane	current
arrivalPos	max
arrivalSpeed	current
line	
personNumber	0
ontainerNumbe	0
departPosLat	center
arrivalPosLat	center
nsertionChecks	all
begin	0.00

Flow attributes

terminate	end
spacing	hsPerHour
end	3600.00
vehsPerHour	500

Parameters

Edit parameters

Netedit attributes

route file	Jnity.rou.xml

Help

Route creator

- Selected edges: 2
- Path edges: 2
- Length: 222.28
- Average speed: 13.89

Finish route creation

Step 2: Run Sumo2Unity integration

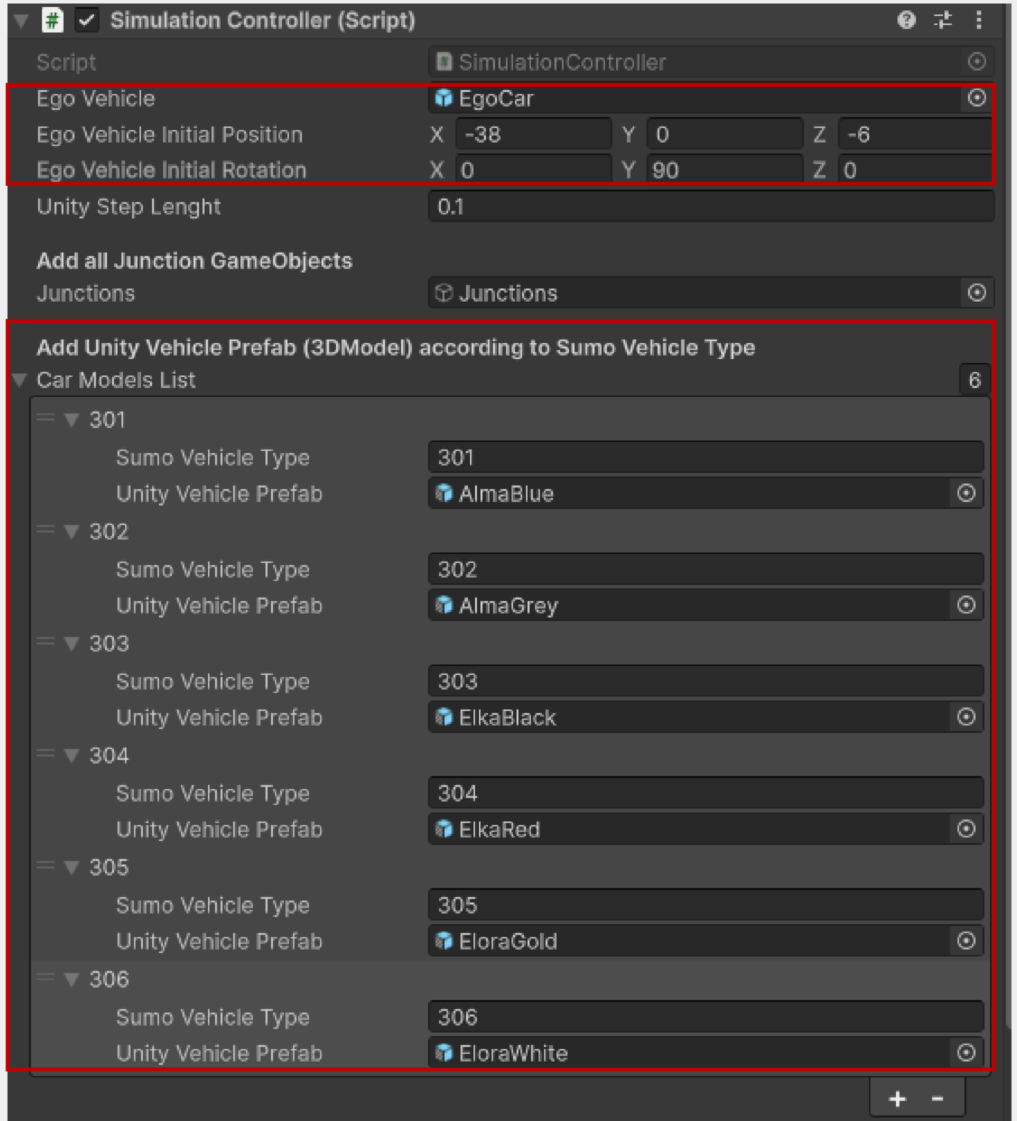
2.1. Unity Steps

C) Assign Ego Vehicle and Traffic Volume in Unity

D) Prepare and Run Python Code (Sumo2Unity.py)

Step 2: Run Sumo2Unity integration

C) Assign Ego Vehicle and Traffic Volume in Unity



Simulation Controller (Script)

Script	SimulationController
Ego Vehicle	EgoCar
Ego Vehicle Initial Position	X -38 Y 0 Z -6
Ego Vehicle Initial Rotation	X 0 Y 90 Z 0
Unity Step Lenght	0.1

Add all Junction GameObjects

Junctions	Junctions

Add Unity Vehicle Prefab (3DModel) according to Sumo Vehicle Type

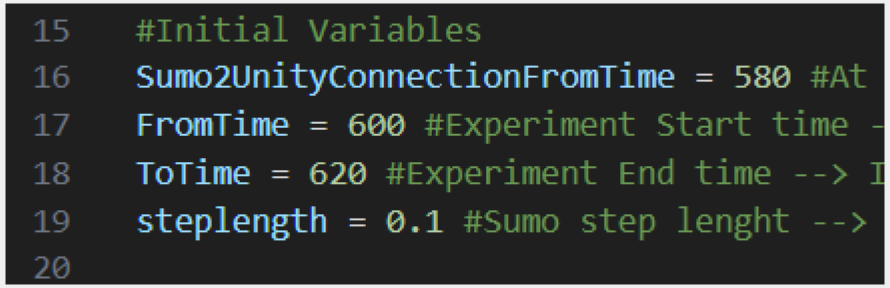
Car Models List 6

	Sumo Vehicle Type	Unity Vehicle Prefab
301	301	AlmaBlue
302	302	AlmaGrey
303	303	ElkaBlack
304	304	ElkaRed
305	305	EloraGold
306	306	EloraWhite

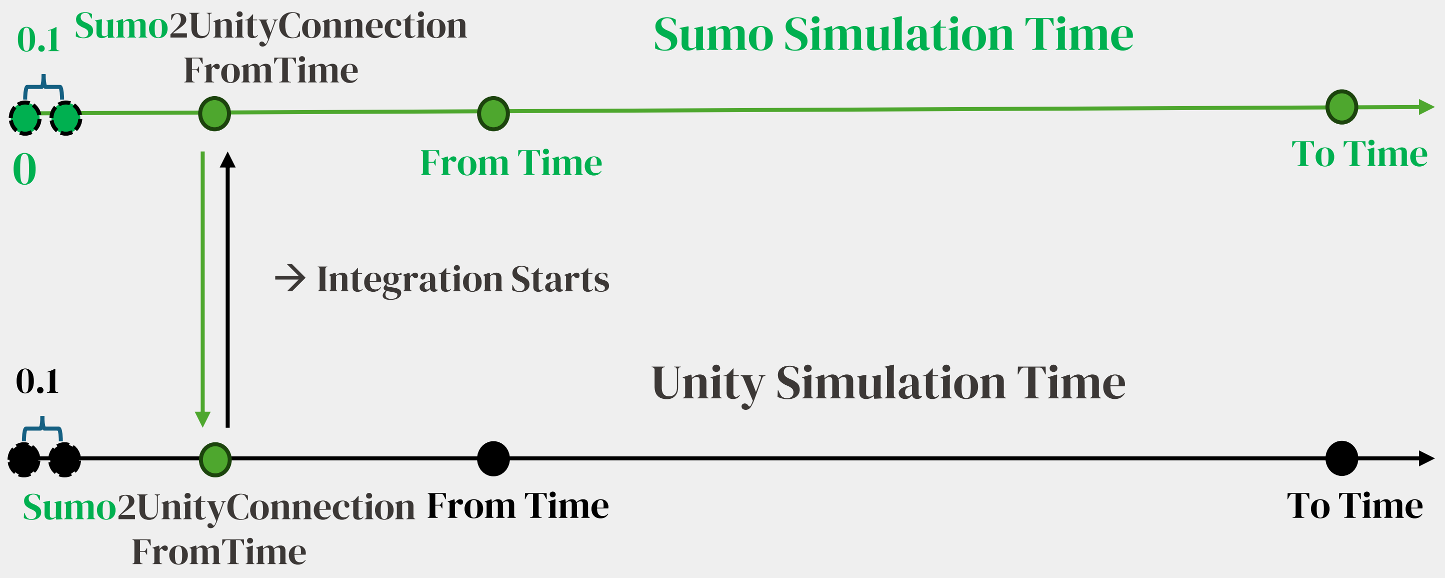
Step 2: Run Sumo2Unity integration

D) Prepare and Run Python Code (Sumo2Unity.py)

- ❑ Initial variables are:
- ❑ Sumo2UnityConnectionFromTime: Integration Start time
- ❑ From Time: Experiment Start time
- ❑ To Time: Experiment End time
- ❑ Step Length:



```
#Initial Variables
Sumo2UnityConnectionFromTime = 580 #At
FromTime = 600 #Experiment Start time -
ToTime = 620 #Experiment End time --> I
steplength = 0.1 #Sumo step lenght -->
```



Step 2: Run Sumo2Unity integration

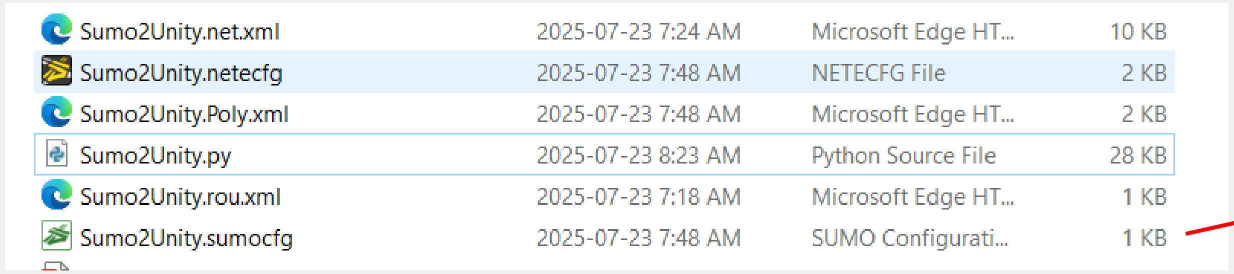
D) Prepare and Run Python Code (Sumo2Unity.py)

- ❑ From time: We normally should give 10 min (600 Seconds for the simulation to run before putting ego vehicle into simulation). This is called warmup period)
- ❑ End time: How long do you want to put the participant in the simulation, for example, if your experiment is 2 min, then end time is 600 +120 second = 780 seconds
- ❑ Step length: is data exchange rate between SUMO and Unity. Default value is 0.1 second. Lower value means more exchanging, and higher accuracy, but it takes a lot of resources. This value should be always equal to Unity Step Length in Unity in Simulation Controller inspector

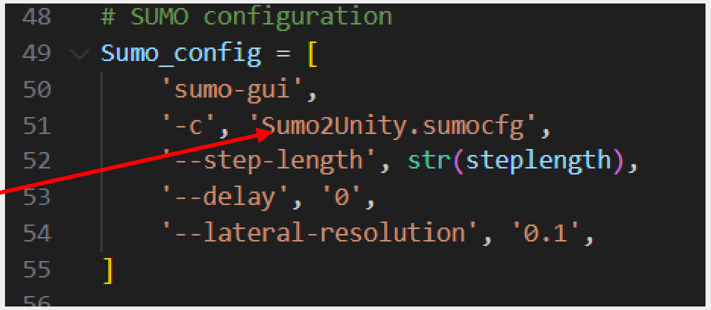
Step 2: Run Sumo2Unity integration

D) Prepare and Run Python Code (Sumo2Unity.py)

- ❑ Make sure you have below files including Sumo2Unity.sumofg in the proper folder “SUMO2Unity\SUMOData”



Sumo2Unity.net.xml	2025-07-23 7:24 AM	Microsoft Edge HT...	10 KB
Sumo2Unity.netecfg	2025-07-23 7:48 AM	NETECFG File	2 KB
Sumo2Unity.Poly.xml	2025-07-23 7:48 AM	Microsoft Edge HT...	2 KB
Sumo2Unity.py	2025-07-23 8:23 AM	Python Source File	28 KB
Sumo2Unity.rou.xml	2025-07-23 7:18 AM	Microsoft Edge HT...	1 KB
Sumo2Unity.sumocfg	2025-07-23 7:48 AM	SUMO Configurati...	1 KB

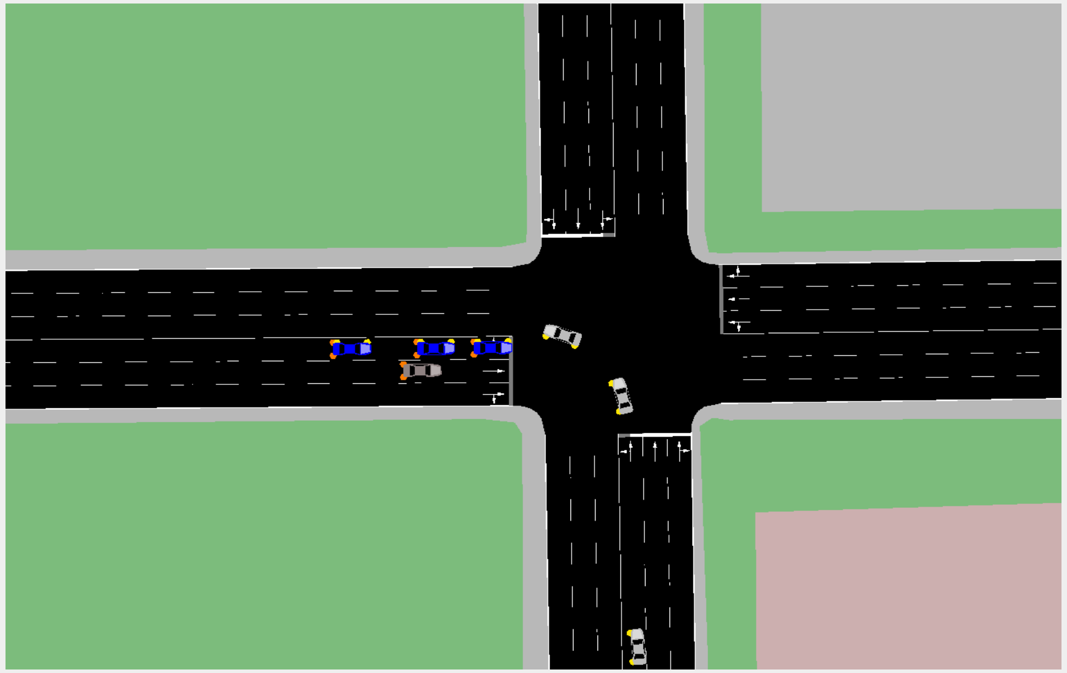


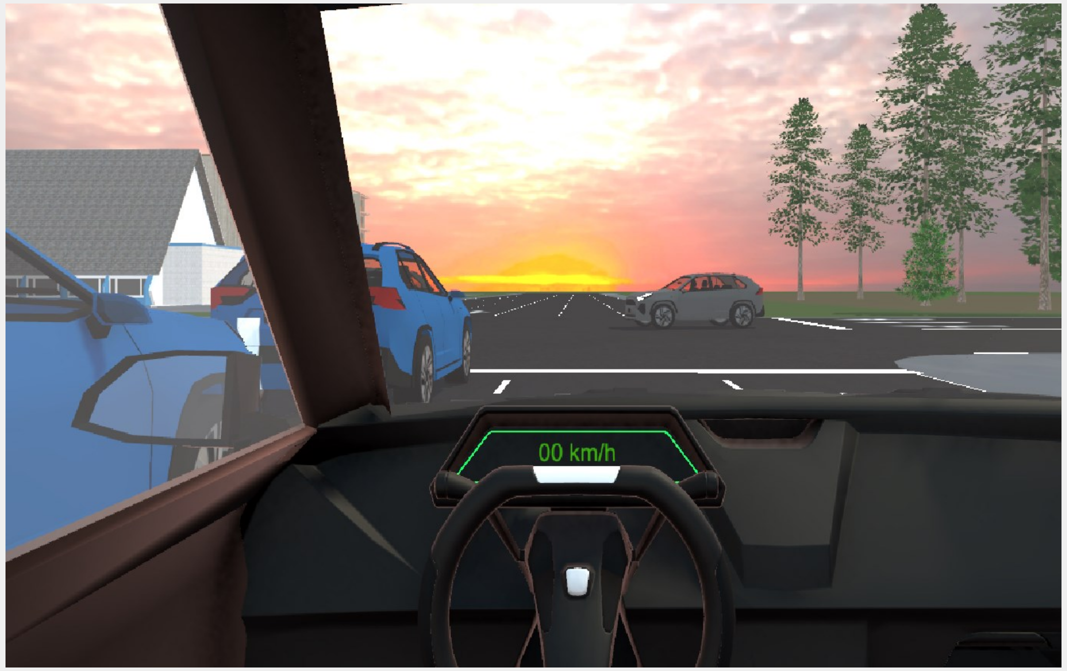
```python
# SUMO configuration
Sumo_config = [
    'sumo-gui',
    '-c', 'Sumo2Unity.sumocfg',
    '--step-length', str(steplength),
    '--delay', '0',
    '--lateral-resolution', '0.1',
]
```

Step 2: Run Sumo2Unity integration

D) Prepare and Run Python Code (Sumo2Unity.py)

- ❑ Run Python
- ❑ When it reaches second 540, SUMO ego car will be added, then Run Unity

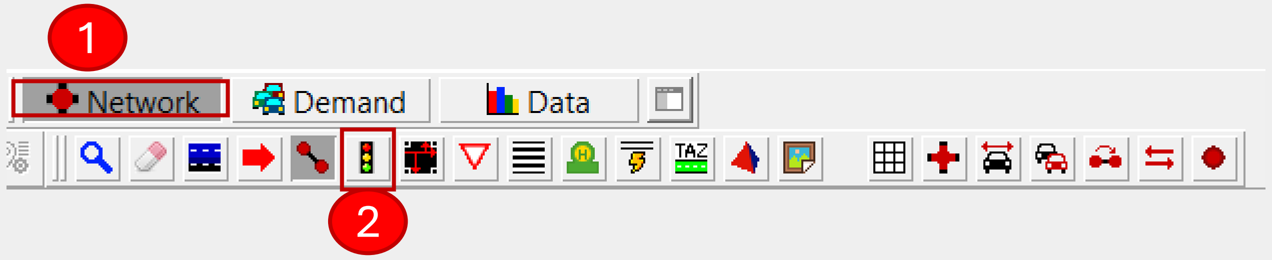


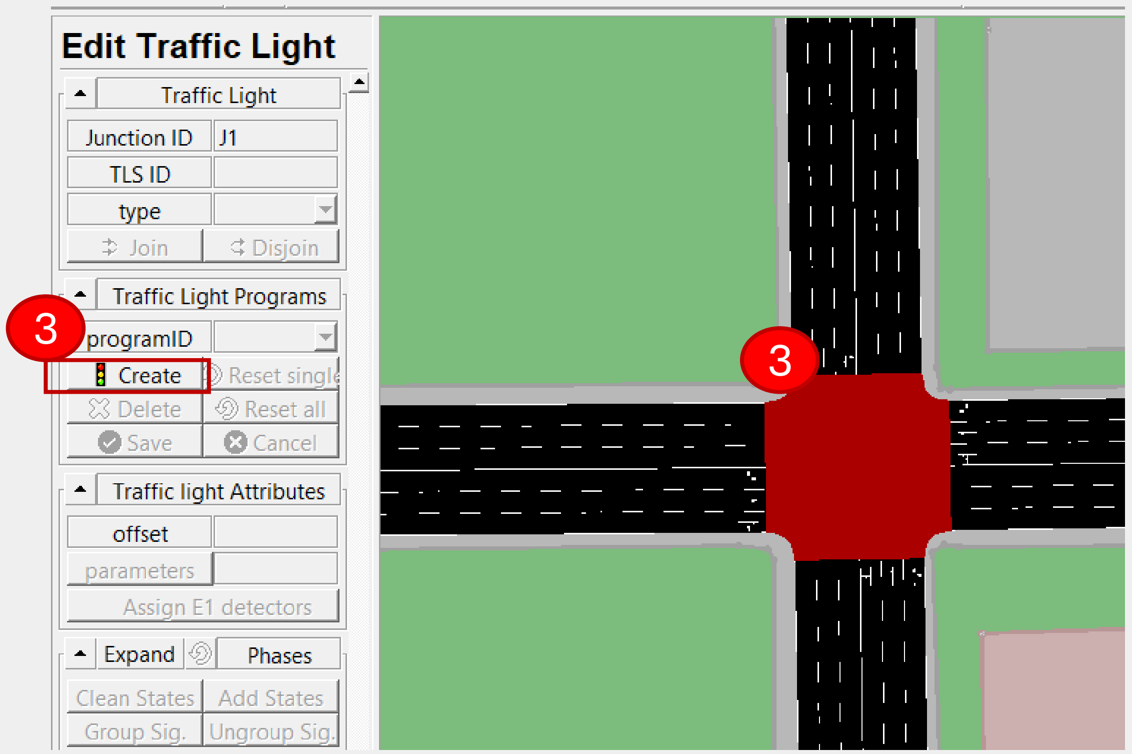


Step 2: Run Sumo2Unity integration

E) Add Traffic Lights in SUMO

- ❑ UI → Select Traffic Light → Select Junction
- ❑ Create
- ❑ File → Save Network



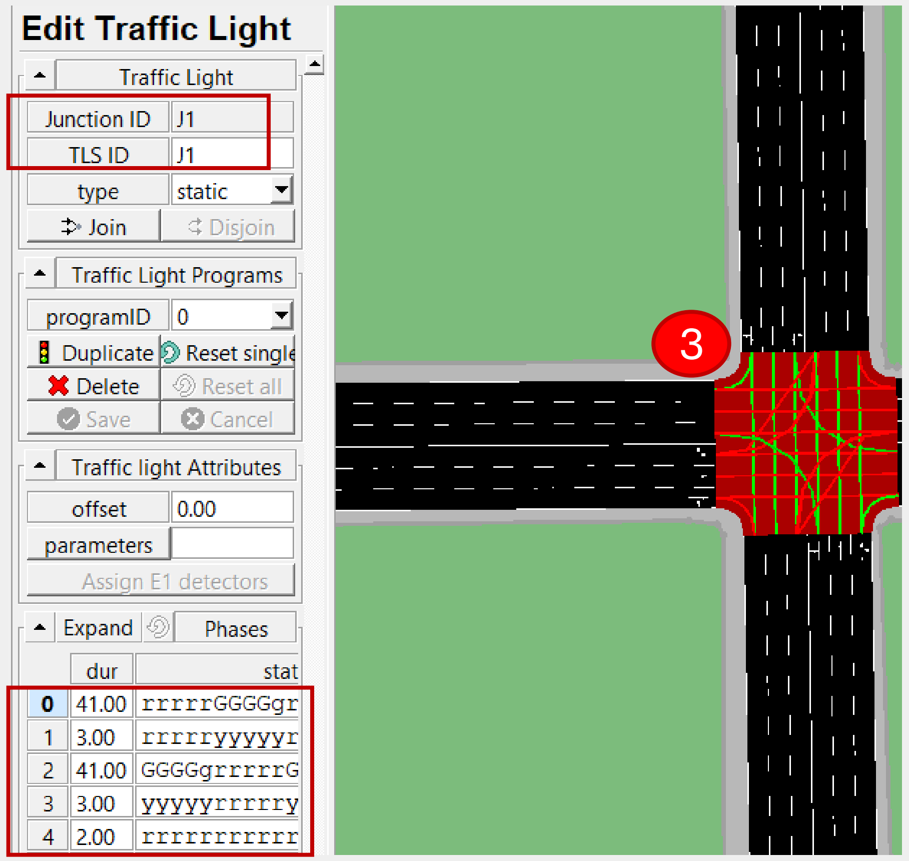


Step 2: Run Sumo2Unity integration

E) Add Traffic Lights in SUMO

- ❑ Junction and TLS ID is J1

- ❑ Explain in next image



Edit Traffic Light

Traffic Light

Junction ID	J1
TLS ID	J1
type	static

Join | Disjoin

Traffic Light Programs

programID	0

Duplicate | Reset single | Delete | Reset all | Save | Cancel

Traffic light Attributes

offset	0.00
parameters	

Assign E1 detectors

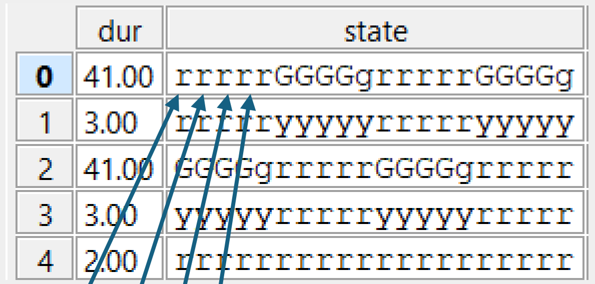
Expand | Phases

	dur	stat
0	41.00	rrrrrGGGGgr
1	3.00	rrrrryyyyyr
2	41.00	GGGGgrrrrrG
3	3.00	yyyyyrrrrry
4	2.00	rrrrrrrrrrr

Step 2: Run Sumo2Unity integration

E) Add Traffic Lights in SUMO

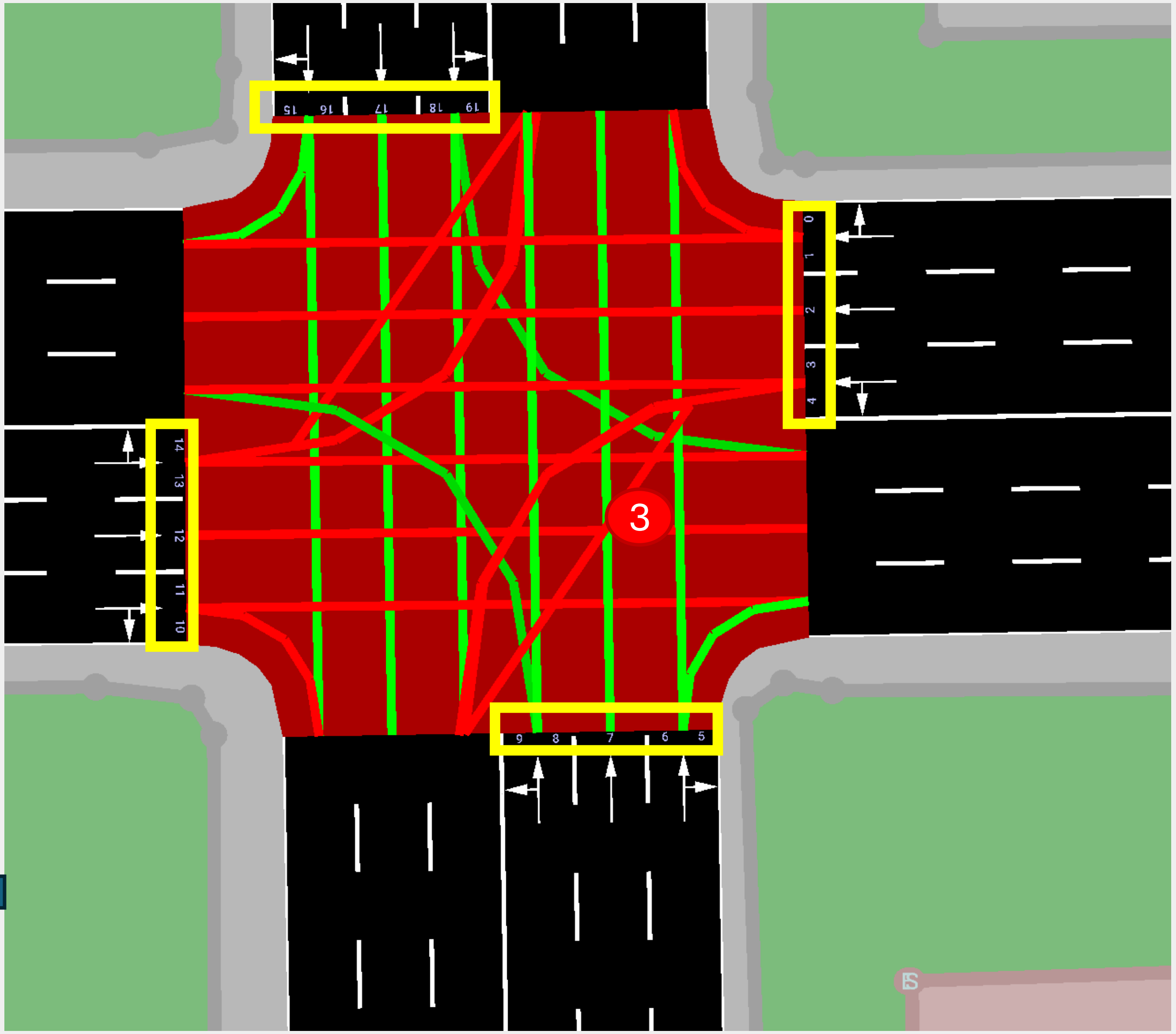
- ❑ Edit → Edit Visualization → junctions → Show Link tls index
- ❑ r : red G:green y: yellow



	dur	state
0	41.00	rrrrrGGGGgrrrrrGGGGg
1	3.00	rrrrryyyyyrrrrryyyyy
2	41.00	GGGGgrrrrrGGGGgrrrrr
3	3.00	yyyyyrrrrryyyyyrrrrr
4	2.00	rrrrrrrrrrrrrrrrrrrr

- ❑ 0, 1, 2, 3, …, 19

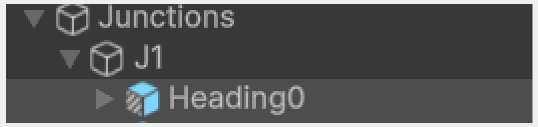
- ❑ See the tutorial video below



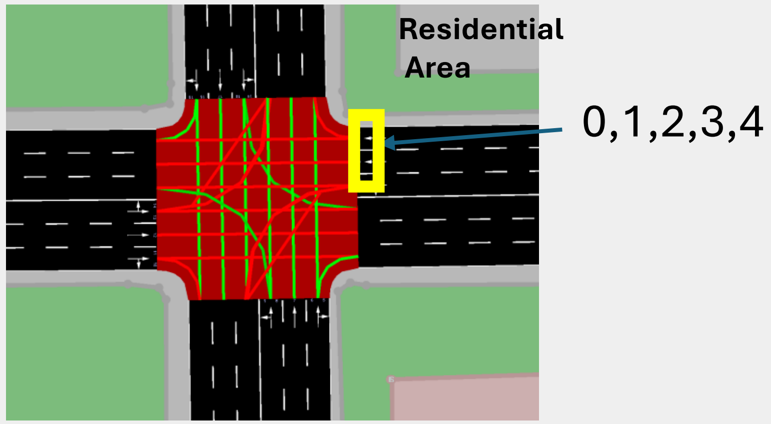
Step 2: Run Sumo2Unity integration

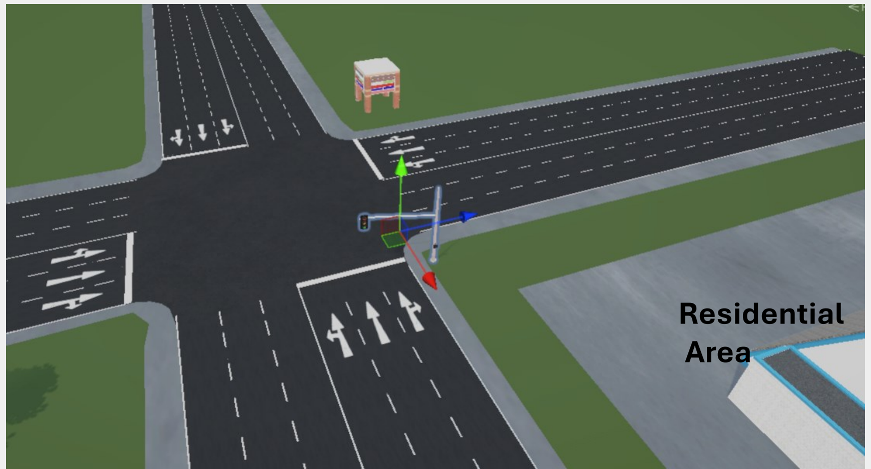
E) Add Traffic Lights in Unity

- ❑ Hierarchy Window → Right Click → Create Empty → name it "Junctions"
- ❑ Junctions → Right Click → Create Empty →Name it "J1" → Move J1 gameObject on top of Junction "J1"
- ❑ Project Window → Resources → Traffic Light → Drag and Drop ThreeLight.prefab into Scene → in Hierarchy Window, put it under gameobject "Junction"



- ❑ See SUMO Traffic Light where the numbers starts from 0 → Locate Traffic Light in Unity there

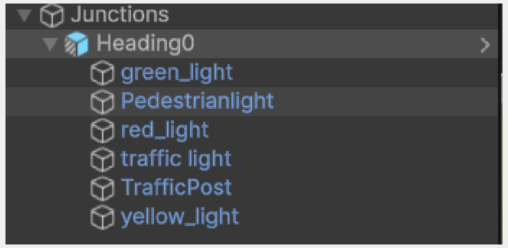




Step 2: Run Sumo2Unity integration

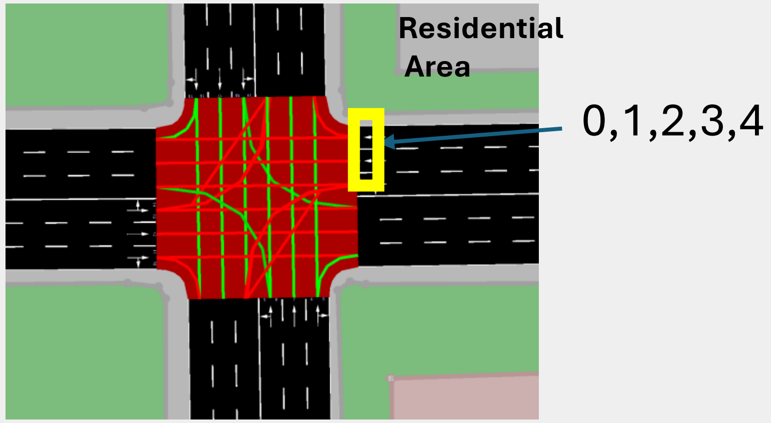
E) Add Traffic Lights in Unity

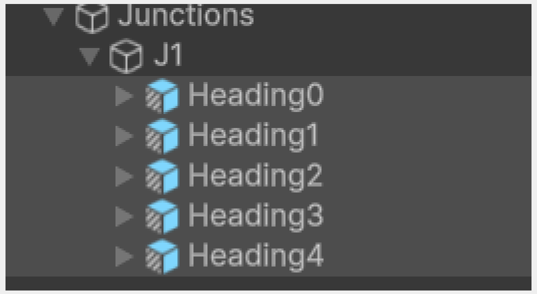
- ❑ Rename it as "Heading0"



Junctions
Heading0
green_light
Pedestrianlight
red_light
traffic light
TrafficPost
yellow_light

- ❑ Duplicate it and name them as "Heading1", "Heading2", "Heading3", "Heading4"
- ❑ See SUMO Traffic Light where the numbers starts from 0 → Locate Traffic Light in Unity there



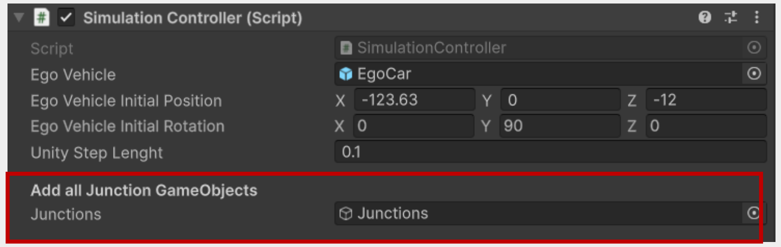


Junctions
J1
Heading0
Heading1
Heading2
Heading3
Heading4

Step 2: Run Sumo2Unity integration

E) Add Traffic Lights in Unity

- ❑ Hierarchy Window → Select Manager → Assign Junction GameObject here
- ❑ Run Python Code and Run Unity
- ❑ The Traffic Light work for head 0-4

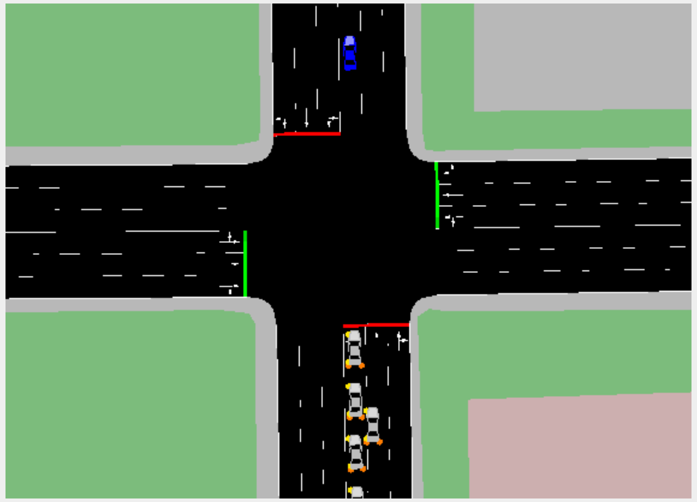


Simulation Controller (Script)

Script	SimulationController
Ego Vehicle	EgoCar
Ego Vehicle Initial Position	X -123.63 Y 0 Z -12
Ego Vehicle Initial Rotation	X 0 Y 90 Z 0
Unity Step Lenght	0.1

Add all Junction GameObjects

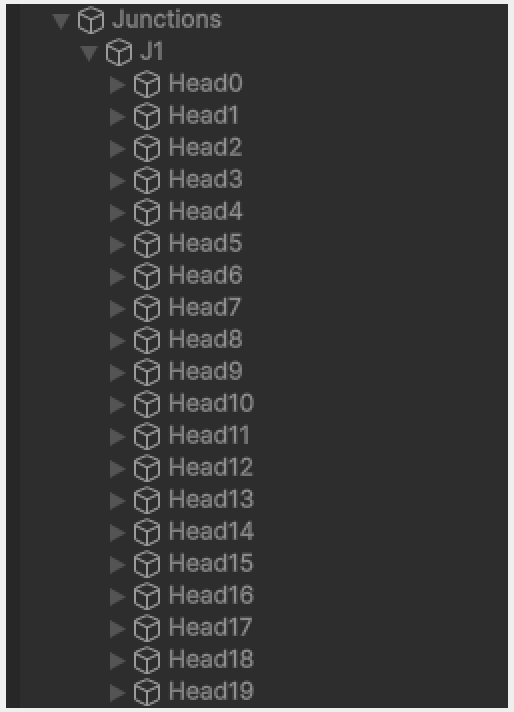
Junctions	Junctions



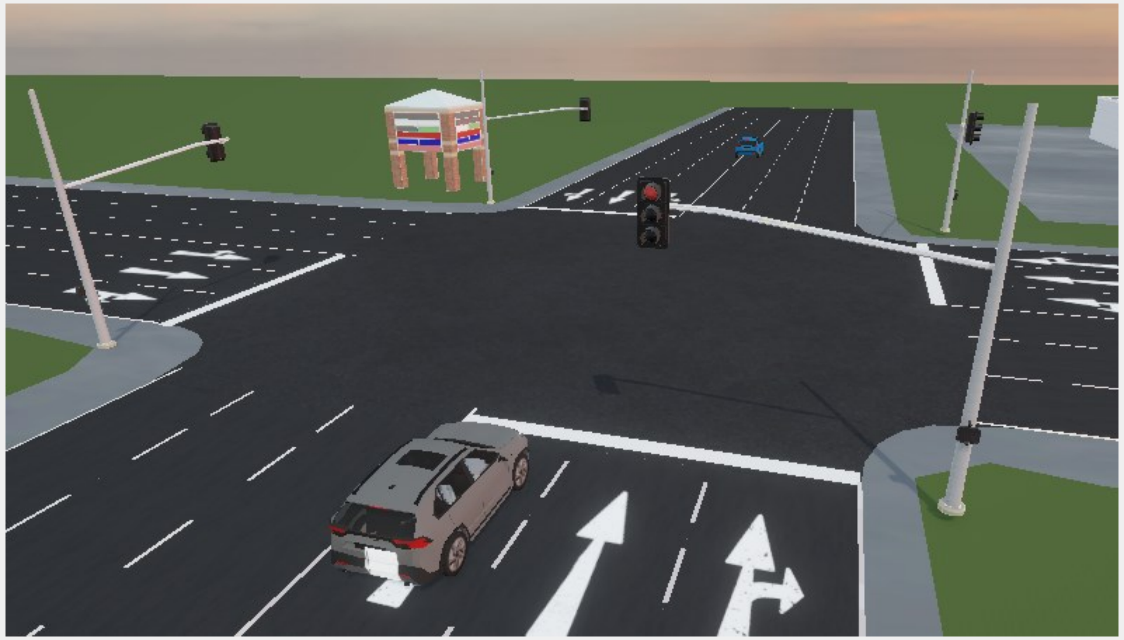
Step 2: Run Sumo2Unity integration

E) Add Traffic Lights in Unity

- ❑ Repeat the same process for Head 5-19
- ❑ Run Python Code and Run Unity
- ❑ Final Result



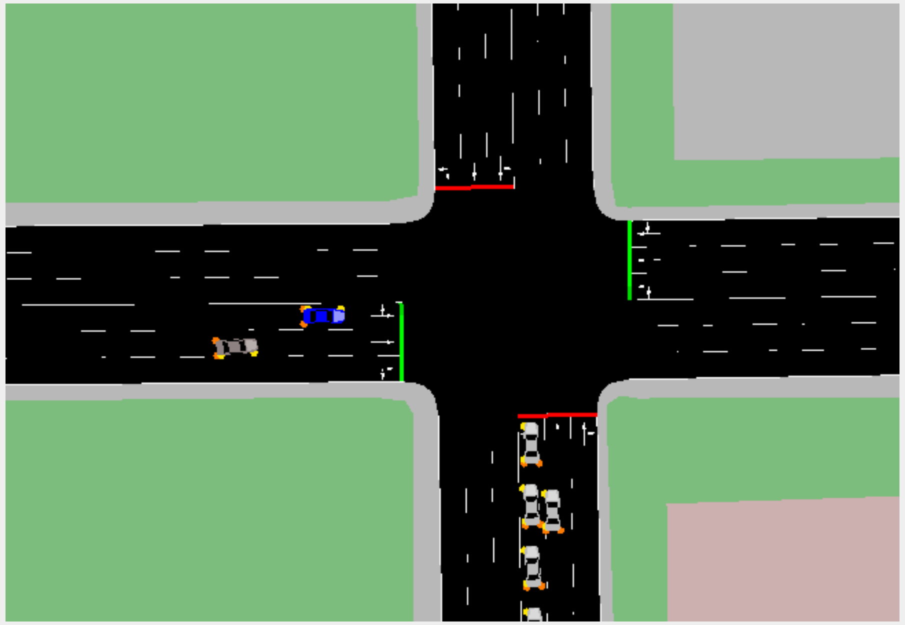
```
▼ Junctions
  ▼ J1
    ▶ Head0
    ▶ Head1
    ▶ Head2
    ▶ Head3
    ▶ Head4
    ▶ Head5
    ▶ Head6
    ▶ Head7
    ▶ Head8
    ▶ Head9
    ▶ Head10
    ▶ Head11
    ▶ Head12
    ▶ Head13
    ▶ Head14
    ▶ Head15
    ▶ Head16
    ▶ Head17
    ▶ Head18
    ▶ Head19
```

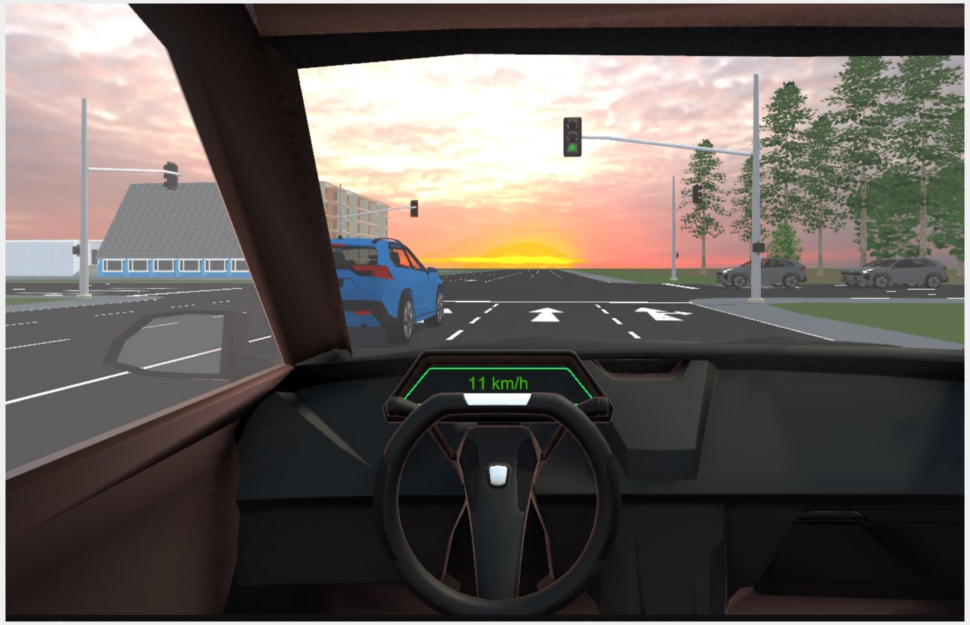


Step 2: Run Sumo2Unity integration

E) Add Traffic Lights in Unity

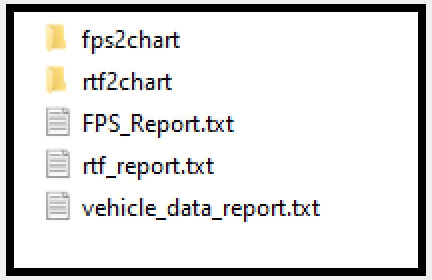
- ❑ Repeat the same process for Head 5-19
- ❑ Run Python Code and Run Unity
- ❑ Final Result





Step 3: Generate Performance Functions

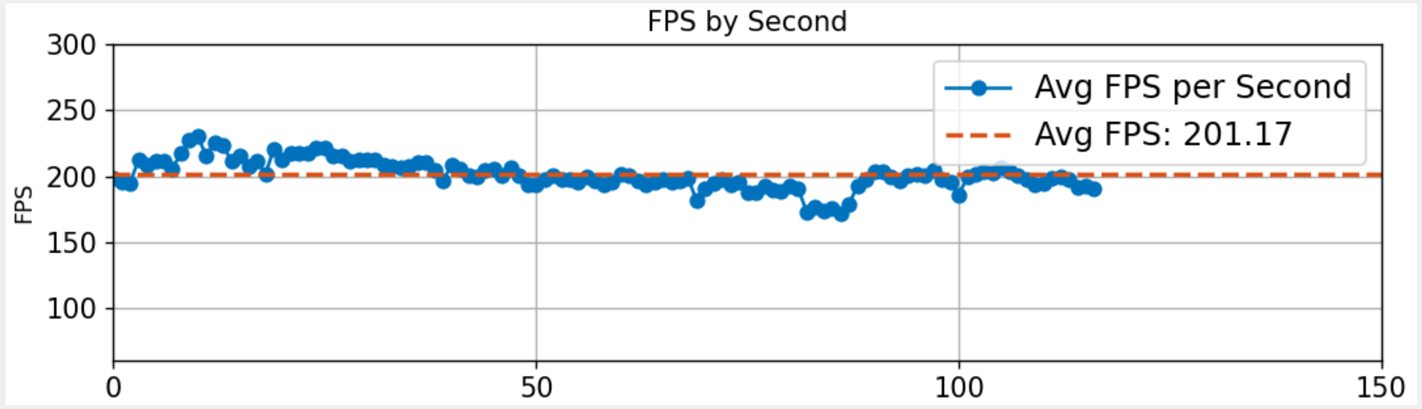
- ❑ **Folder Results**



fps2chart
rtf2chart
FPS_Report.txt
rtf_report.txt
vehicle_data_report.txt

- ❑ Copy and Paste FPS_Report.txt → Folder “fps2chart” → Replace with “FPS_Report.txt”

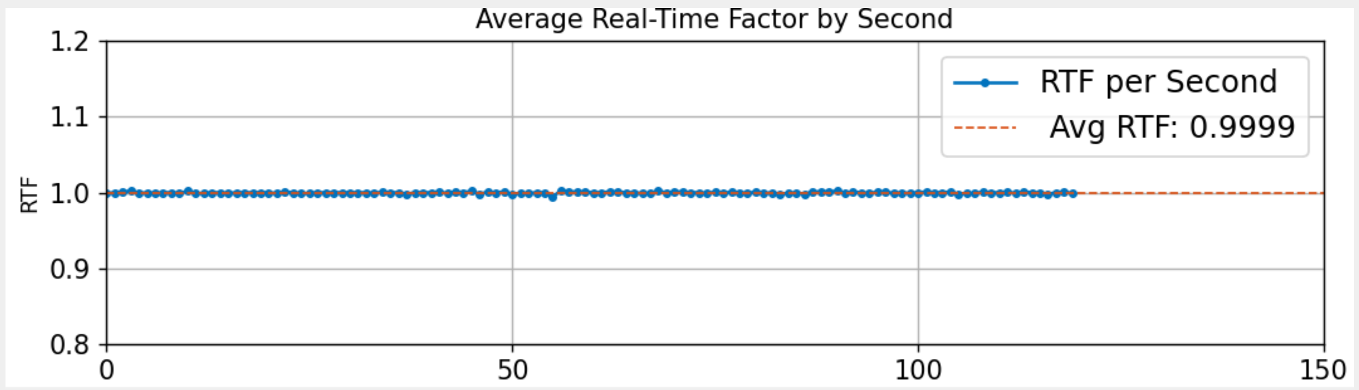
- ❑ Run fps2chart.py →



Step 3: Generate Performance Functions

- ❑ Copy and Paste rtf_report.txt → Folder “rtf2chart” → Replace with “rtf_report.txt”

- ❑ Run rtf2chart.py →


Average Real-Time Factor by Second
RTF
RTF per Second
Avg RTF: 0.9999