

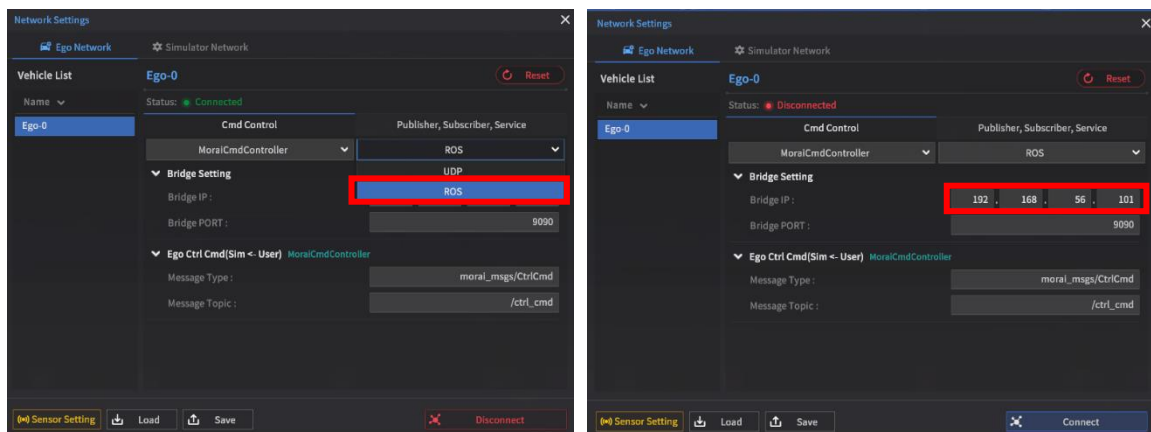
## 공통

### □ 우분투 터미널

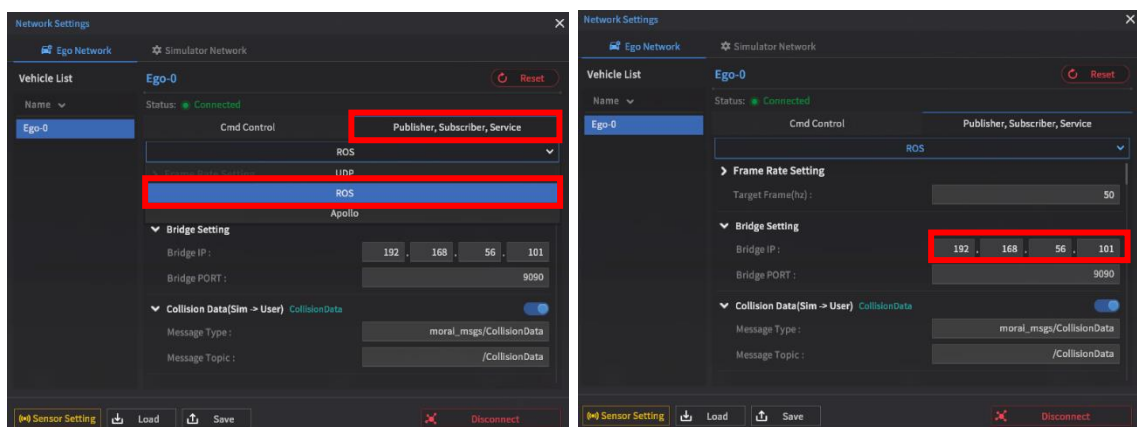
웹소켓 서버와 연결시켜줍니다.

```
roslaunch rosbridge_server rosbridge_websocket.launch
```

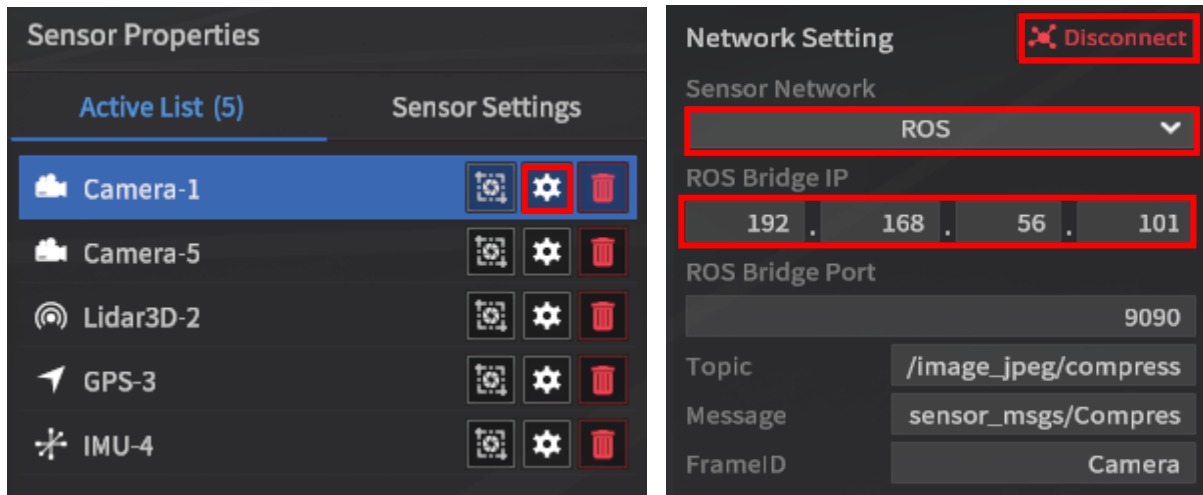
### □ MORAI 시뮬레이터



1. "F4"를 눌러 "Network Settings" 옵션을 띄웁니다
2. "Ego Network CmdControl"에서 ROS 선택 후 Bridge IP 192.168.56.101 연결



3. "Ego Network Publisher, Subscriber, Service"에서 "ROS"를 선택 후 Bridge IP 192.168.56.101 연결



4. "F3"을 눌러 센서 리스트에서 톱니바퀴 버튼을 눌러 ROS 선택 후 Bridge IP 192.168.56.101 연결

## 시연 1 (Dijkstra)

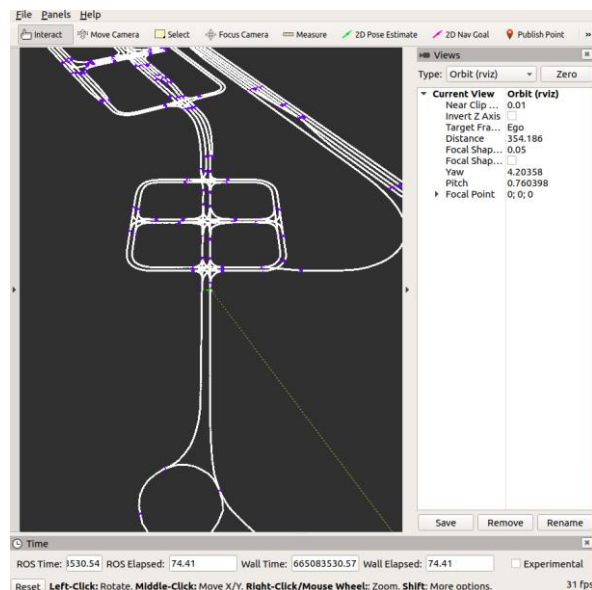
### □ 우분투 터미널

출발점과 도착점을 설정 후 각 노드의 최단경로를 생성해 주행하는 경우

새로운 우분투 터미널에

```
roslaunch control dijkstra_driving.launch
```

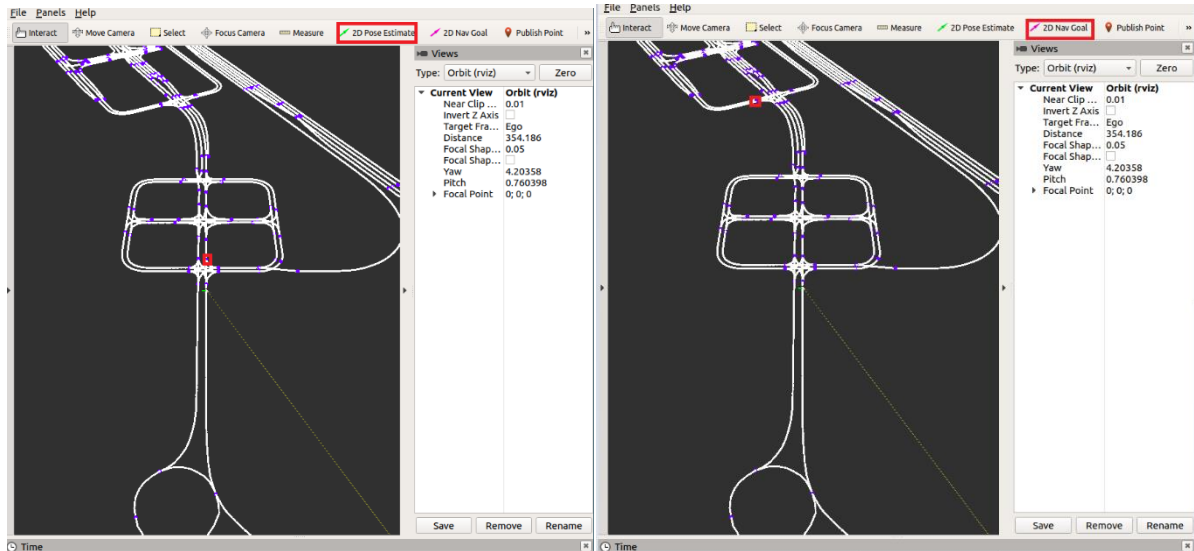
터미널에 코드를 입력하면 다음과 같은 rviz 창이 나타납니다.



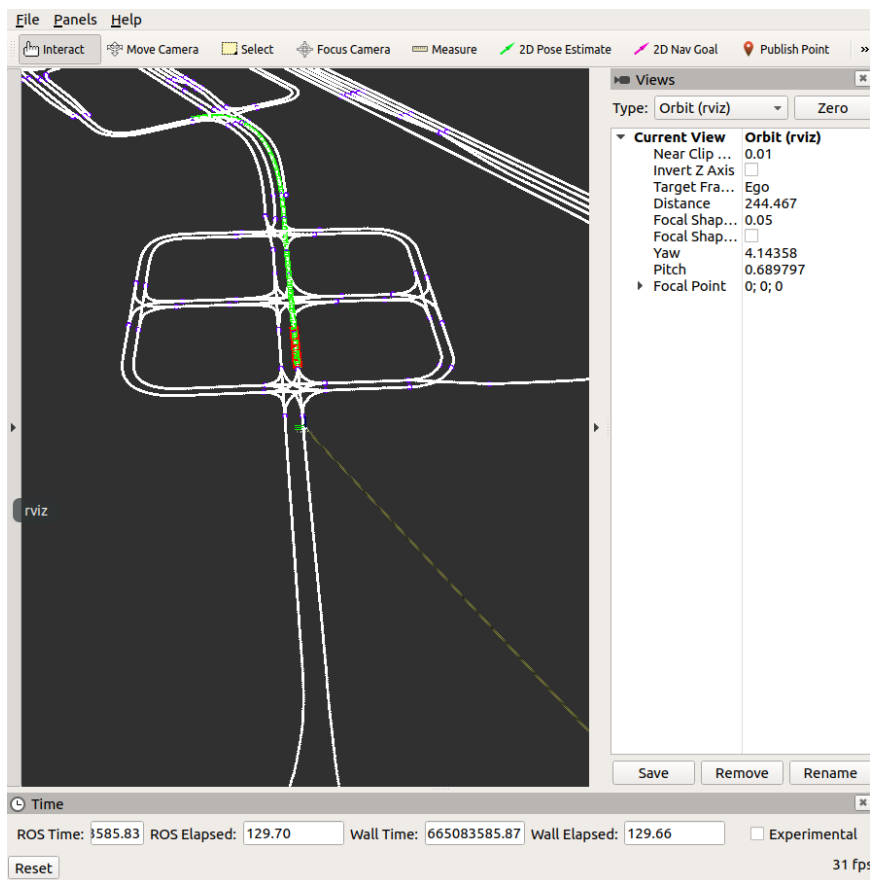
이 rviz 창에서 출발점과 도착점을 정해줍니다.

2D Pose Estimate를 클릭하고 보라색 노드들 중 출발점으로 하고 싶은 노드를 클릭합니다

출발 노드를 정했다면 2D Nav Goal을 클릭하고 노드들 중 도착지점으로 하고 싶은 노드를 클릭합니다.



도착 노드도 정했다면 dijkstra 알고리즘으로 두 노드 사이의 경로가 초록색으로 생성됩니다.

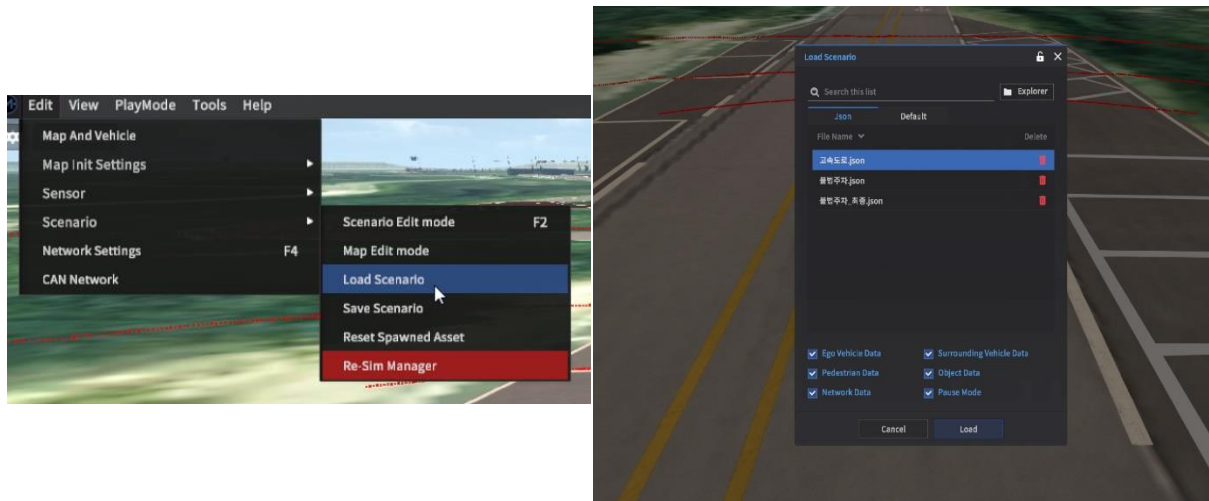


## 시연 2 (고속도로 차선변경)

### □ MORAI 시뮬레이터

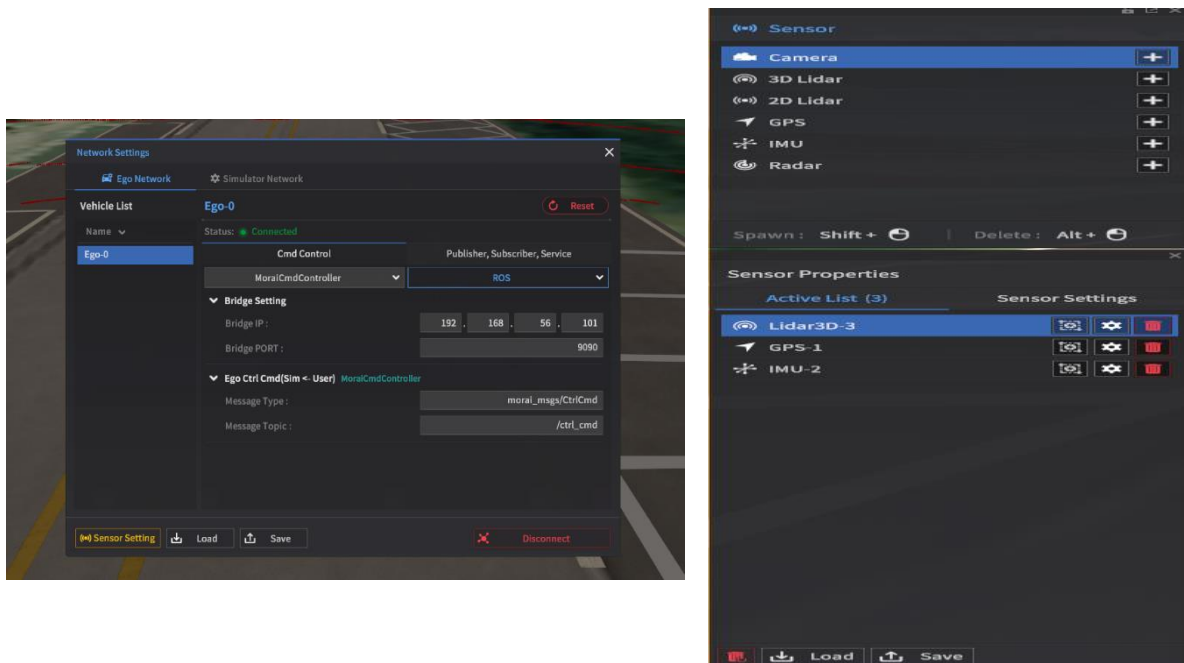
고속도로에서 차량 충돌 방지를 위해 차선 변경하는 경우

시뮬레이터에서 Edit – Scenario – Load scenario 를 선택해 고속도로.json 파일을 불러옵니다.



그러면 저장되었던 고속도로와 차량배치 시나리오가 나타납니다.

F4 와 f3을 눌러 공통 부분의 네트워크, 센서 네트워크 세팅을 합니다.



## □ 우분투 터미널

새로운 우분투 터미널에서

```
roslaunch control kcity_highway.launch
```

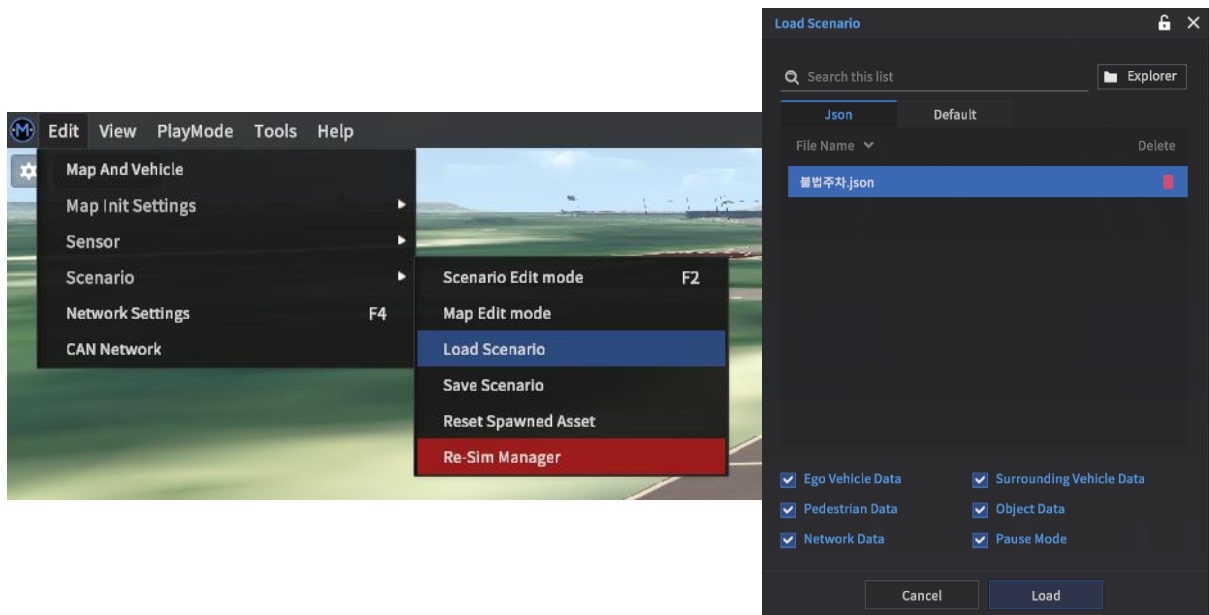
를 실행시켜줍니다.

실행이 되었다면 Q를 눌러 auto 모드를 실행시키면 생성된 경로를 따라 자율주행 합니다.

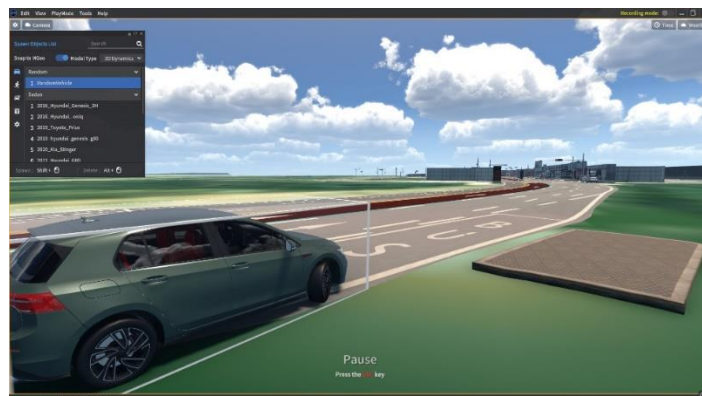
## 시연 3 (불법 주차 차량 인식)

### □ MORAI 시뮬레이터

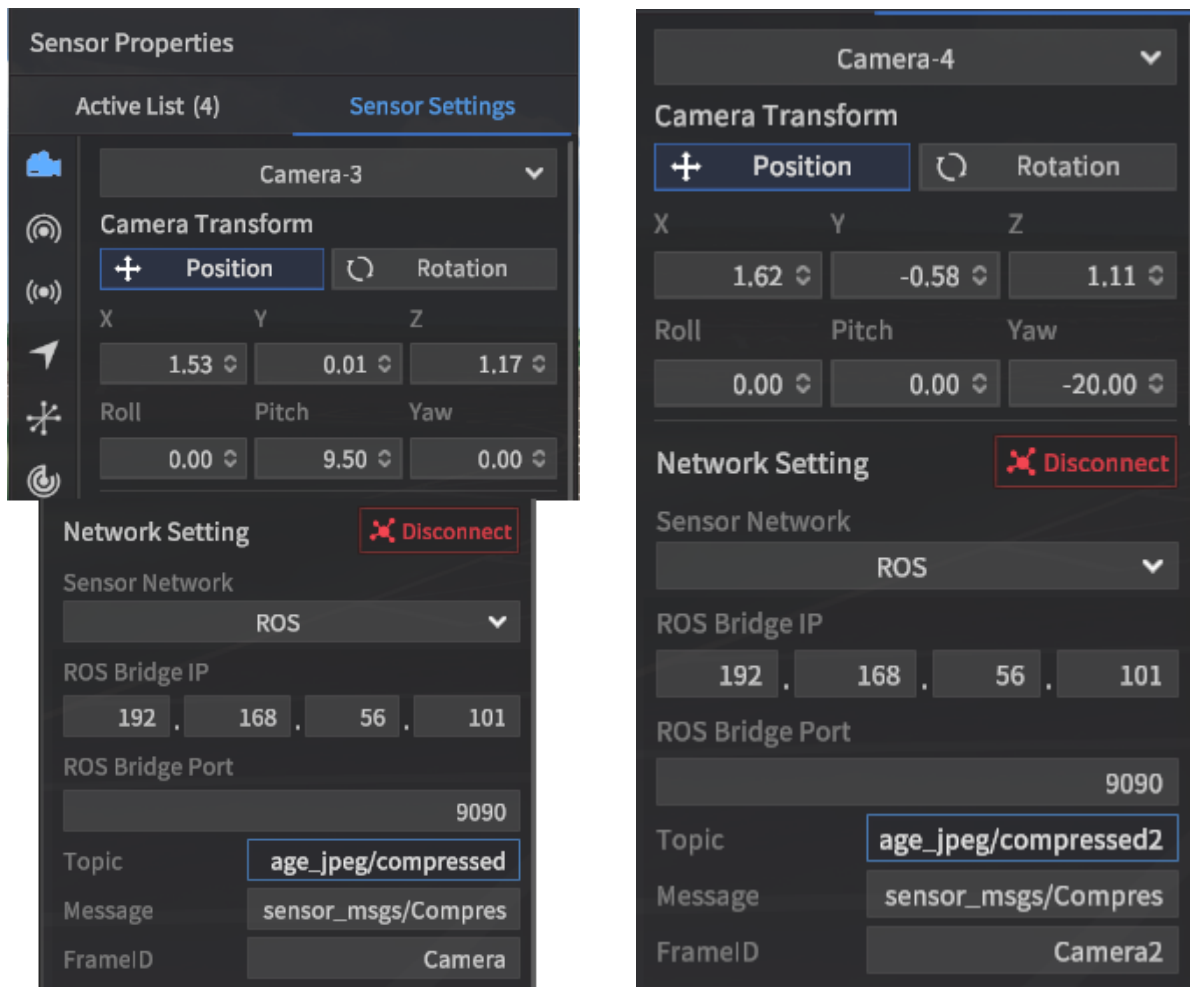
시뮬레이터의 edit - scenario - load scenario 를 선택 해 불법주차.json 파일을 불러옵니다



그러면 저장되었던 차량배치 시나리오가 나타납니다.



F4와 F3을 눌러 공통부분의 네트워크, 센서 네트워크를 설정합니다. (카메라 2개 세팅)



## □ 우분투 터미널

새로운 우분투 터미널에

```
roslaunch control kcity_noparking.launch
```

을 실행합니다.

실행이 되었다면 q를 눌러 auto mode로 실행시켜 주행을 확인합니다.

mysql 실행 DB에 불법주차 차량 정보를 확인합니다.

```
mysql> select * from tb_car_illegal;
```

car_num image_data	time	npc_x	npc_y	prior_cnt
2016_Hyundai_Santafe https://gotothemars.s3.ap-northeast-2.amazonaws.com/car_picture/2016_Hyundai_Santafe.jpg	2022-10-07 04:26:48	147.7579345703125	1424.3232421875	2
2019_Hyundai_Nexo https://gotothemars.s3.ap-northeast-2.amazonaws.com/car_picture/2019_Hyundai_Nexo.jpg	2022-10-07 04:26:42	142.94186401367188	1349.604248046875	2
2020_Kia_Stinger https://gotothemars.s3.ap-northeast-2.amazonaws.com/car_picture/2020_Kia_Stinger.jpg	2022-10-07 04:26:56	147.0850372314453	1540.827880859375	2
2021_Volkswagen_Golf_GTI https://gotothemars.s3.ap-northeast-2.amazonaws.com/car_picture/2021_Volkswagen_Golf_GTI.jpg	2022-10-07 04:28:45	116.85871124267578	1274.74609375	3

4 rows in set (0.00 sec)

## ※ 불법주차 알고리즘

1. 먼저 MySQL과 S3를 연결합니다.
2. 만약 Ego와 타 차량 간의 거리가 특정 거리만큼 떨어져 있고 특정 각도를 만족한다면 위반한 차량에 대한 정보를 쓸 것입니다. 언제 걸렸는지(time), 어디에 주차되어 있었는지(npc\_x, npc\_y), 몇 번 위반되었는지(prior\_cnt), 차량 사진(image\_data) 등을 저장할 것입니다.
3. 만약 위반한 전과가 있다면 time, npc\_x, npc\_y를 새로 업데이트하고 prior\_cnt를 1 증가시켜 업데이트합니다.
4. 실질적인 차량 사진 data는 S3에 저장되어 있습니다. MySQL은 이에 대한 주소가 저장됩니다.