LGSVL SIMULATOR

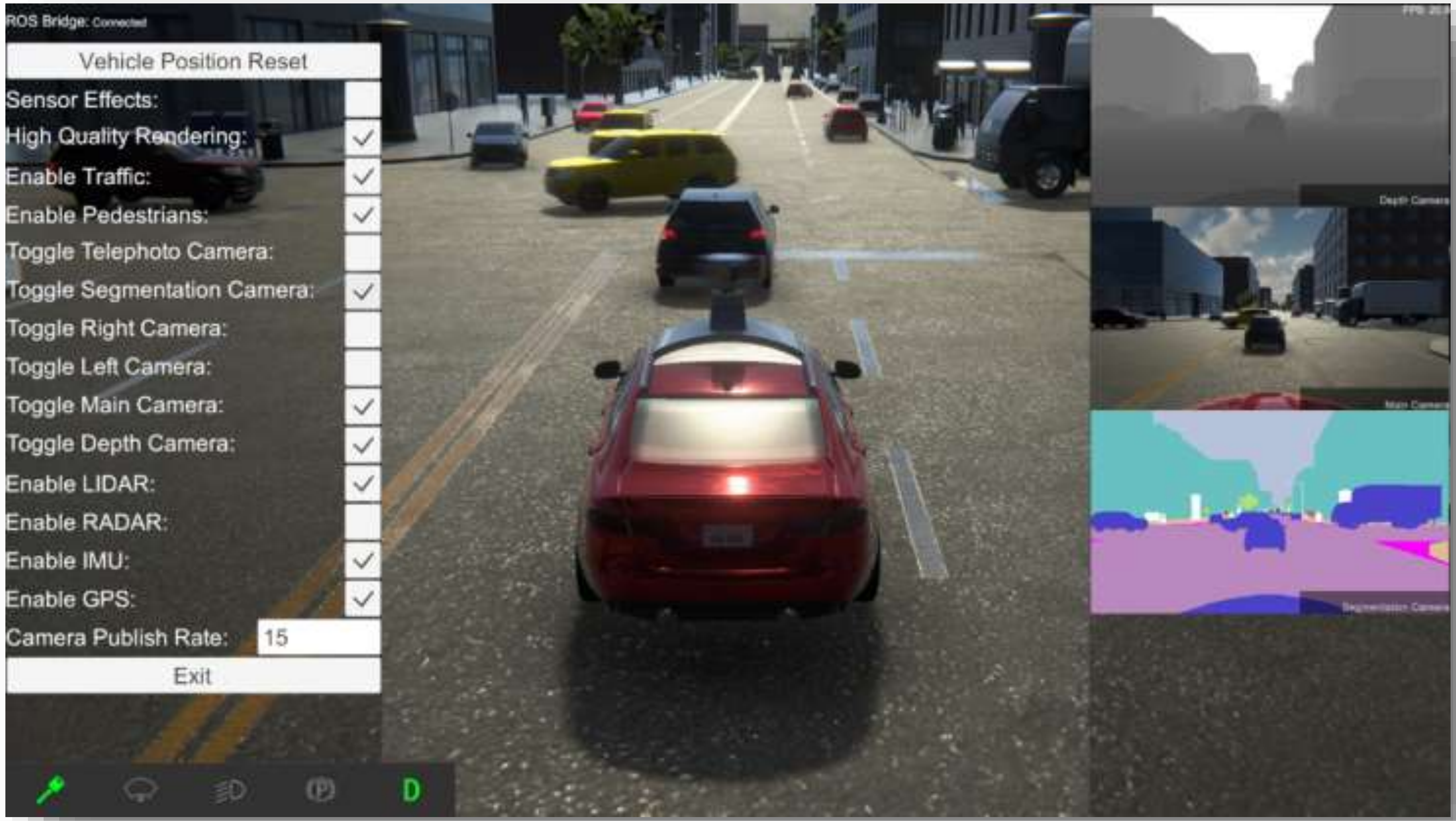
April 2019 Update

Website: https://www.lgsvlsimulator.com
Contact: contact@lgsvlsimulator.com
GitHub: https://github.com/lgsvl/simulator

LGSVL Autonomous Driving Simulator

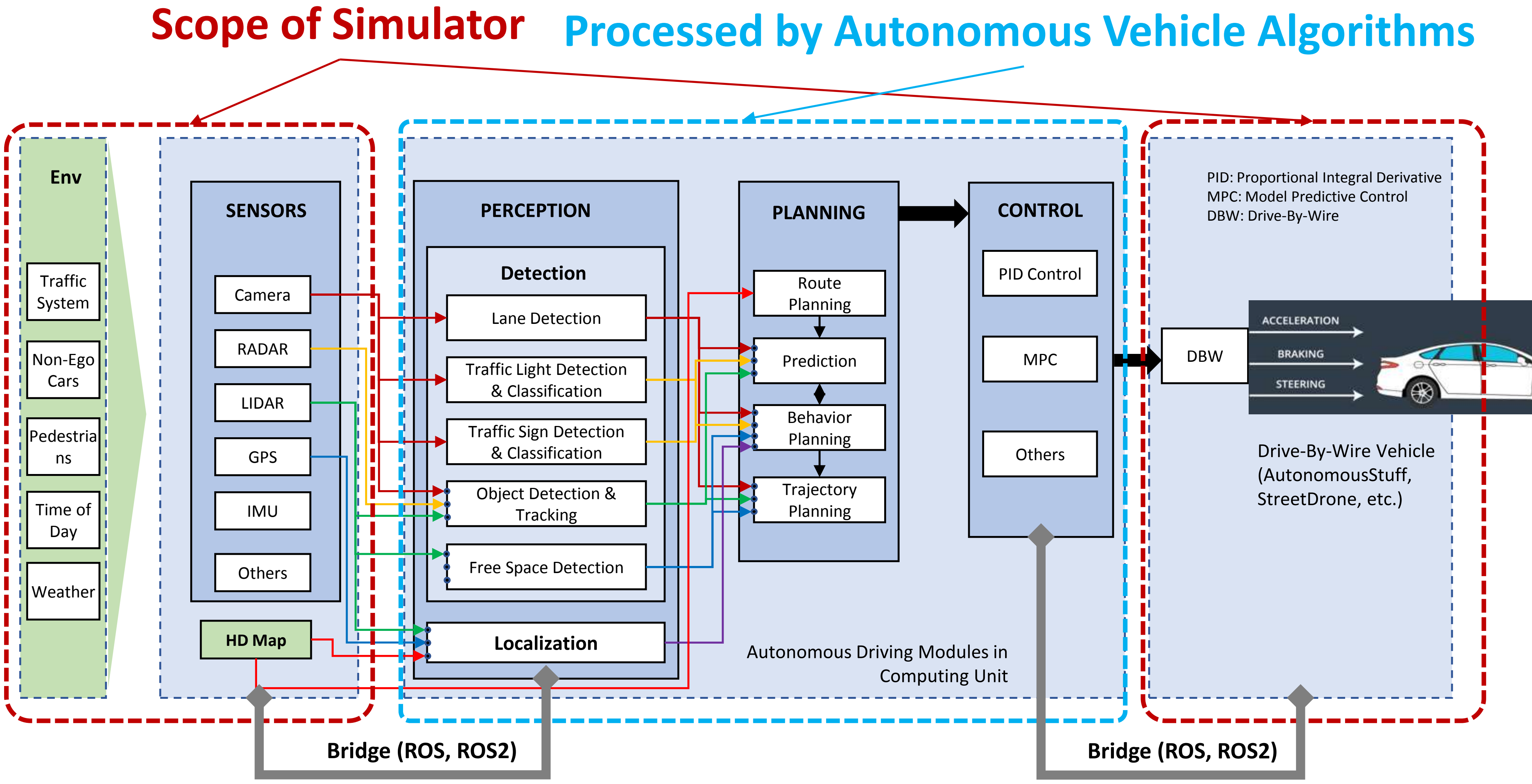
The LGSVL Simulator is a simulator that facilitates testing and development of autonomous driving software systems. It enables developers to simulate billions of miles and arbitrary edge case scenarios to speed up algorithm development and system integration.

The simulator is under active development on Github, and we hope to grow our developer community to best serve the autonomous vehicle and robotics industries in accelerating the testing and validation of autonomous systems.



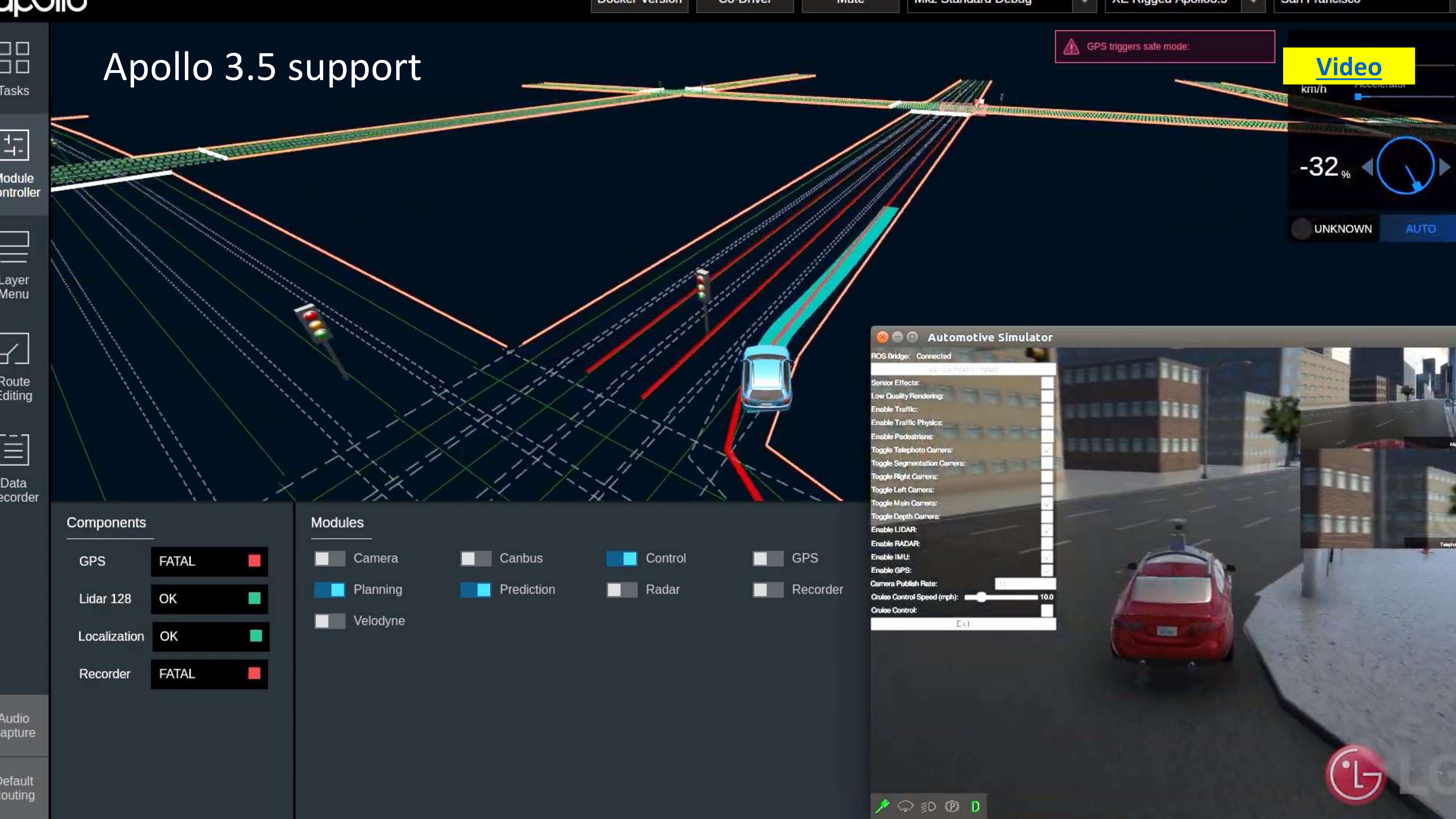
Sensor Effects: ☐
High Quality Rendering: ✓
Enable Traffic: ✓
Enable Pedestrians: ✓
Toggle Telephoto Camera: ☐
Toggle Segmentation Camera: ✓
Toggle Right Camera: ☐
Toggle Left Camera: ☐
Toggle Main Camera: ✓
Toggle Depth Camera: ✓
Enable LIDAR: ✓
Enable RADAR: ☐
Enable IMU: ✓
Enable GPS: ✓
Camera Publish Rate: 15

AD Simulation – Architectural Overview



Apollo 3.5 support

Video



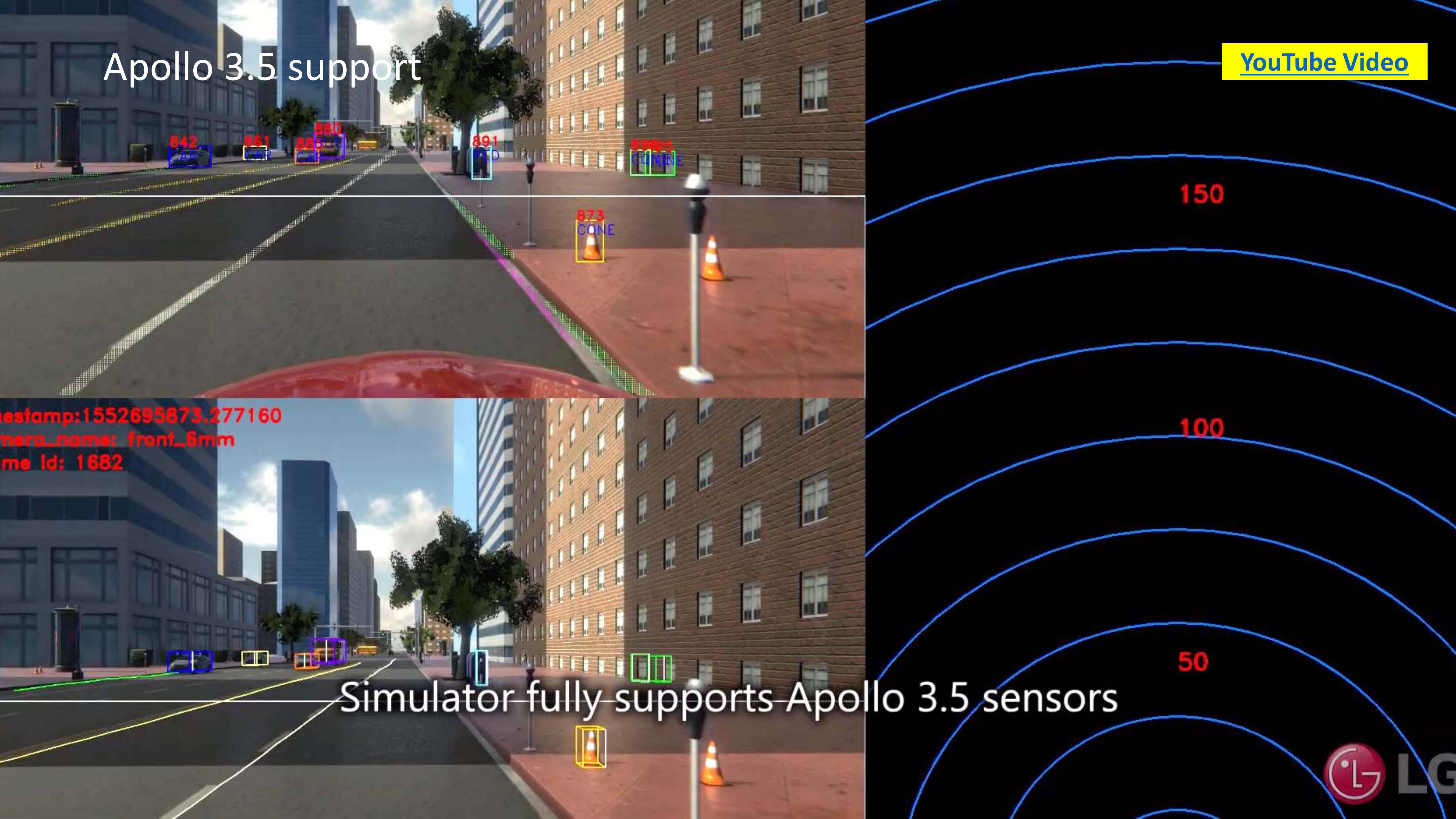
Components	
GPS	FATAL 🟥
Lidar 128	OK 🟩
Localization	OK 🟩
Recorder	FATAL 🟥

ROS Bridge: Connected

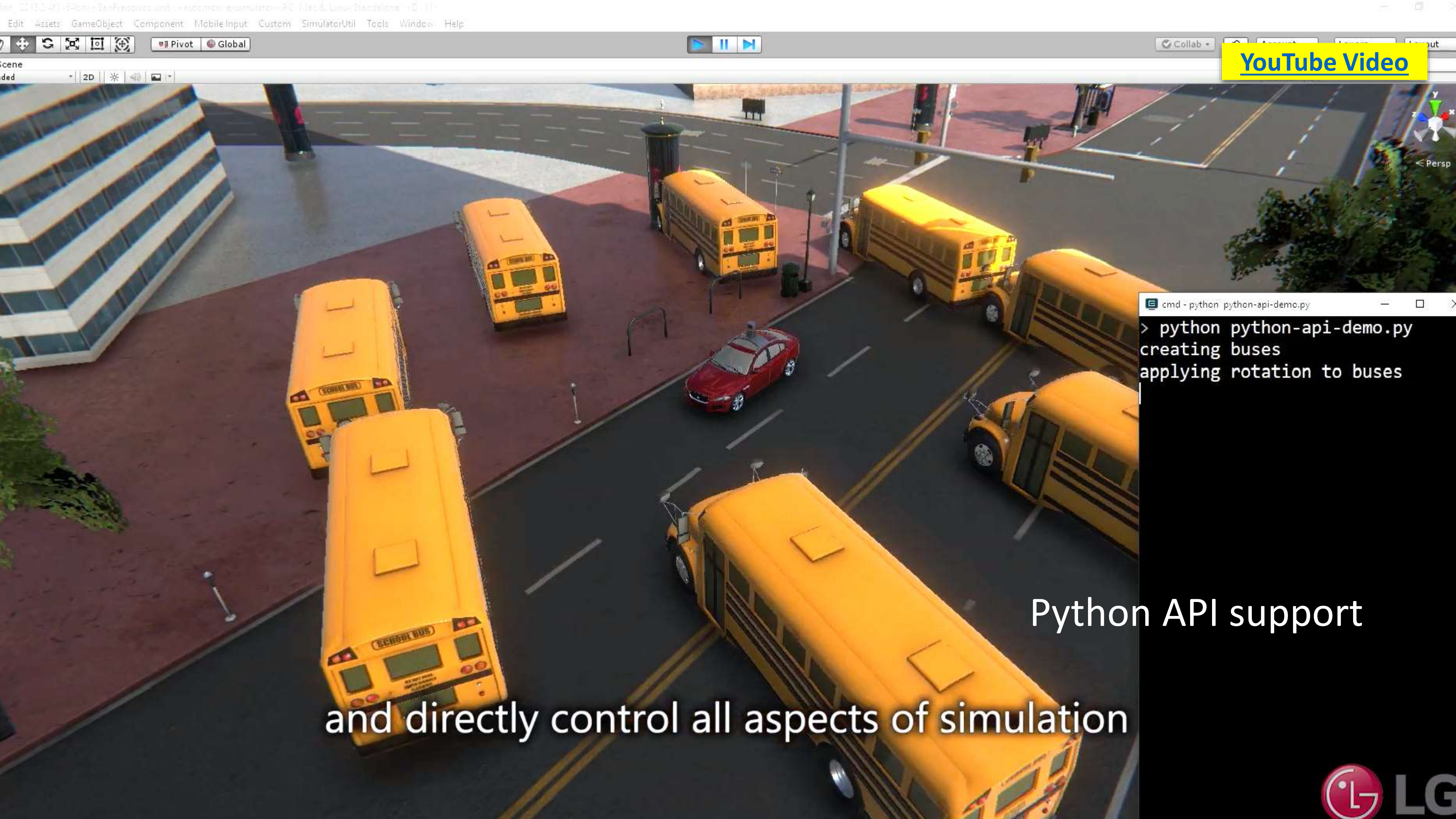
Sensor Effects:	
Low Quality Rendering:	
Enable Traffic:	
Enable Traffic Physics:	
Enable Pedestrians:	
Toggle Telephoto Camera:	✓
Toggle Segmentation Camera:	
Toggle Right Camera:	
Toggle Left Camera:	
Toggle Main Camera:	✓
Toggle Depth Camera:	
Enable LIDAR:	✓
Enable RADAR:	
Enable IMU:	✓
Enable GPS:	✓
Camera Publish Rate:	15
Cruise Control Speed (mph):	10.0
Cruise Control:	

Apollo 3.5 support

YouTube Video



Simulator fully supports Apollo 3.5 sensors

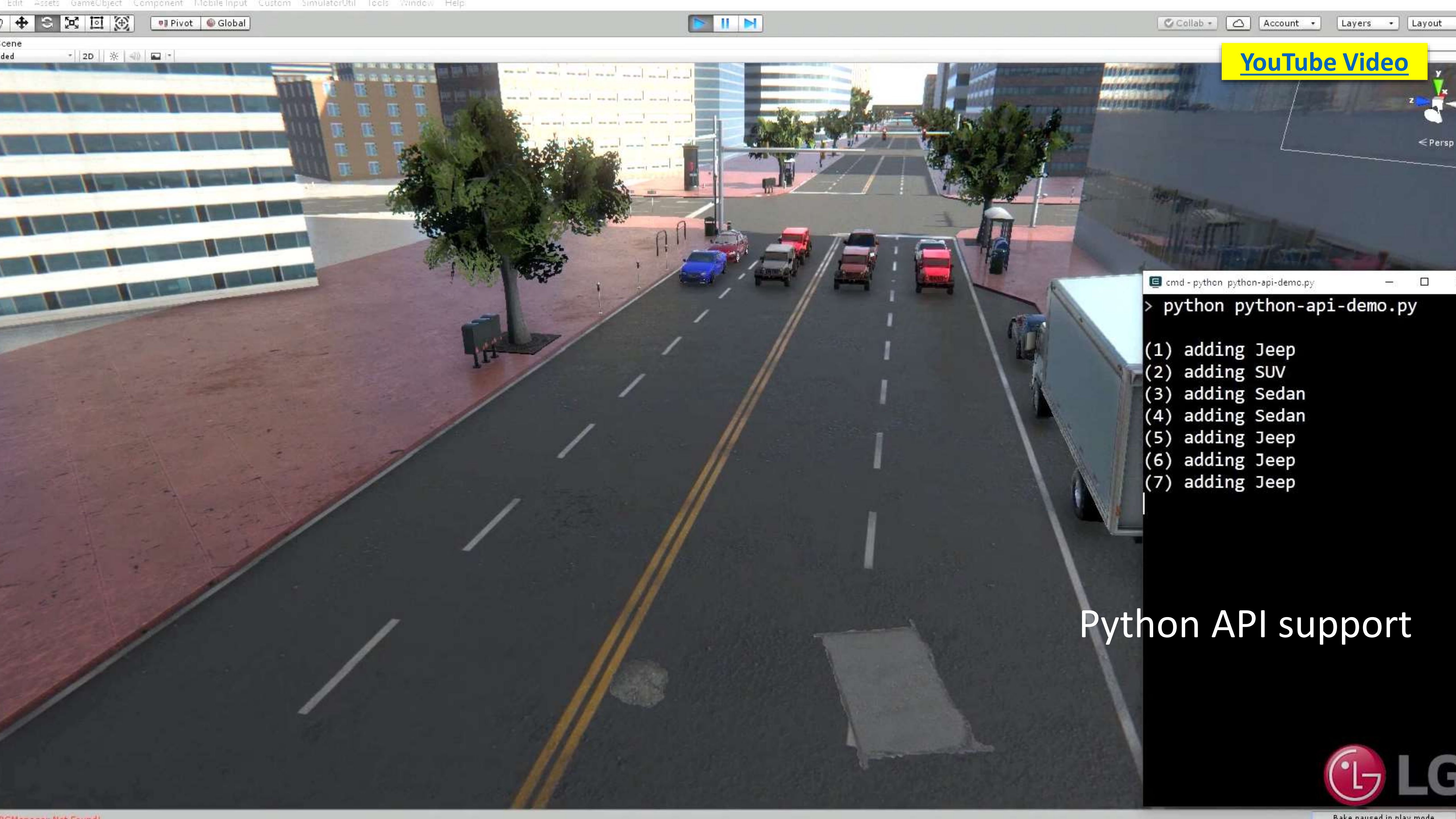

Edit Assets GameObject Component Mobile Input Custom SimulatorUtil Tools Window Help
Pivot
Global
Collab
Scene
2D
Persp
SCHOOL BUS
cmd - python python-api-demo.py

YouTube Video

```
> python python-api-demo.py
creating buses
applying rotation to buses
```

Python API support

and directly control all aspects of simulation



YouTube Video

```
cmd - python  python-api-demo.py
> python python-api-demo.py

(1) adding Jeep
(2) adding SUV
(3) adding Sedan
(4) adding Sedan
(5) adding Jeep
(6) adding Jeep
(7) adding Jeep
```

Python API support

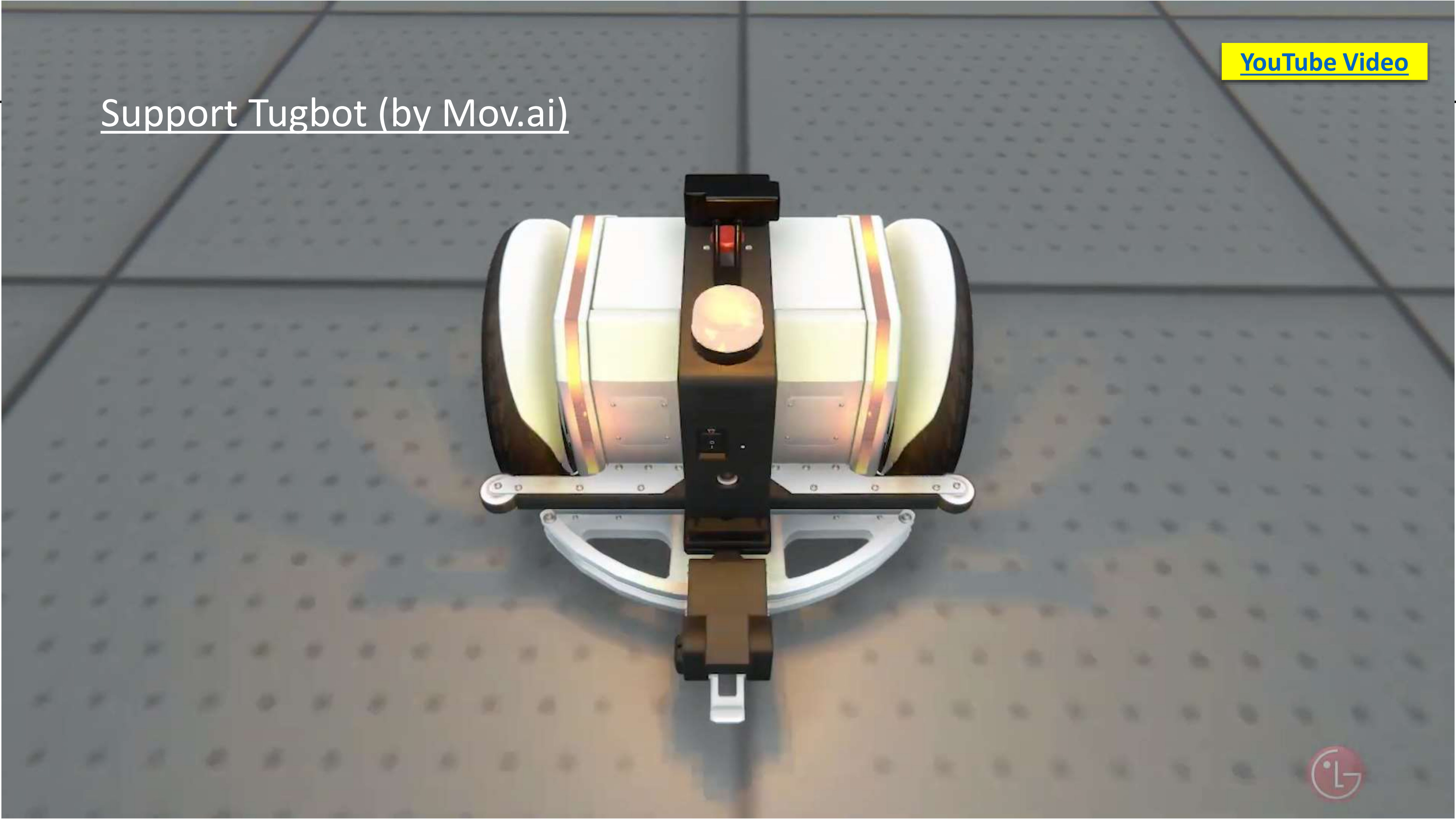
New Akihabara (Tokyo) Map

YouTube Video

MENT
Complete integration took less than one day
LG

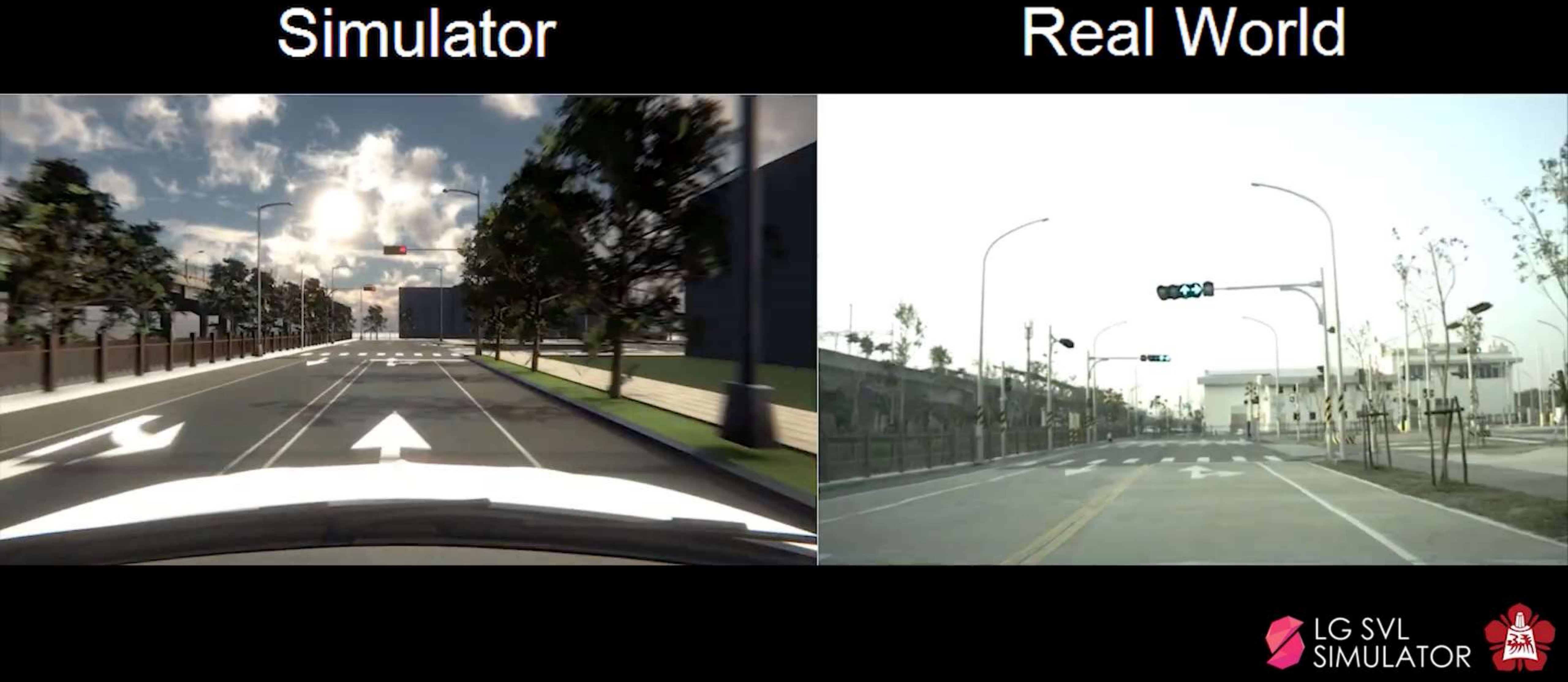
Support Tugbot (by Mov.ai)

YouTube Video

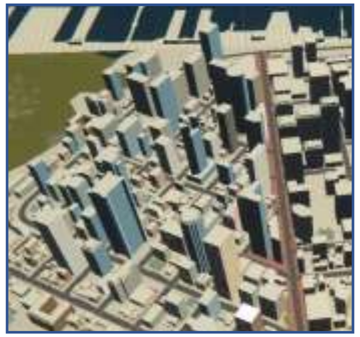


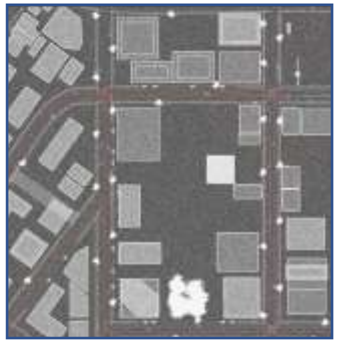
Digital Twin of Taiwan Car LAB

YouTube Video



LGSVL Simulator - Features

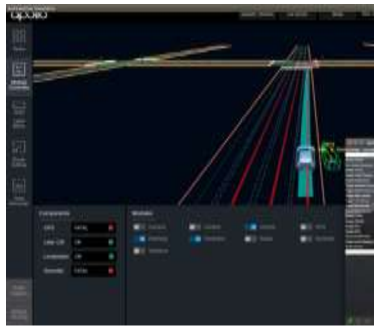

Maps

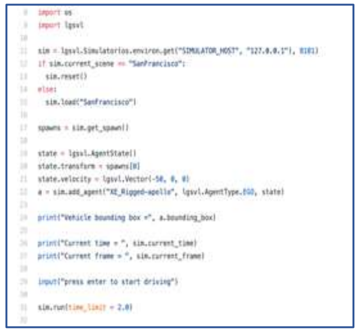

Map Annotation Tool

Vehicle models with sensor & vehicle physics model

Ground Truth Information

ROS/ROS2 Support

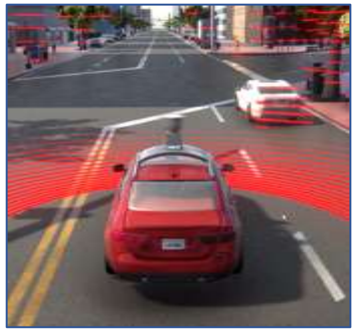

Open Source AD Stack support (Baidu Apollo & Autoware)



```
import os
import lgsvl

sim = lgsvl.Simulator(os.environ.get("SIMULATOR_HOST", "127.0.0.1"), 8181)
if sim.current_scene == "SanFrancisco":
  sim.reset()
else:
  sim.load("SanFrancisco")

spawns = sim.get_spawn()

state = lgsvl.AgentState()
state.transform = spawns[0]
state.velocity = lgsvl.Vector(-50, 0, 0)
a = sim.add_agent("XE_Rigged-apollo", lgsvl.AgentType.EGO, state)

print("Vehicle bounding box =", a.bounding_box)

print("Current time = ", sim.current_time)
print("Current frame = ", sim.current_frame)

input("press enter to start driving")

sim.run(time_limit = 2.0)
```

APIs

Different NPC Models & Traffic system (Cars, Pedestrians, Bikes, etc.)


Sensor Simulations (LiDAR, Radar, GPU, IMU, etc.)

Weather & Time of day

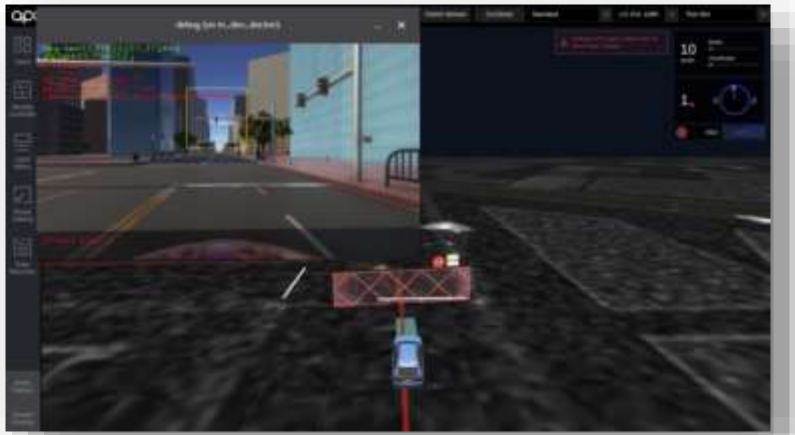
HDRP

Synthetic data generation for deep learning

Features - Open Source AD Stack Support

Currently, supports Baidu Apollo 3.0 and 3.5 and Autoware by default.
Any custom AD stacks can be supported as well.

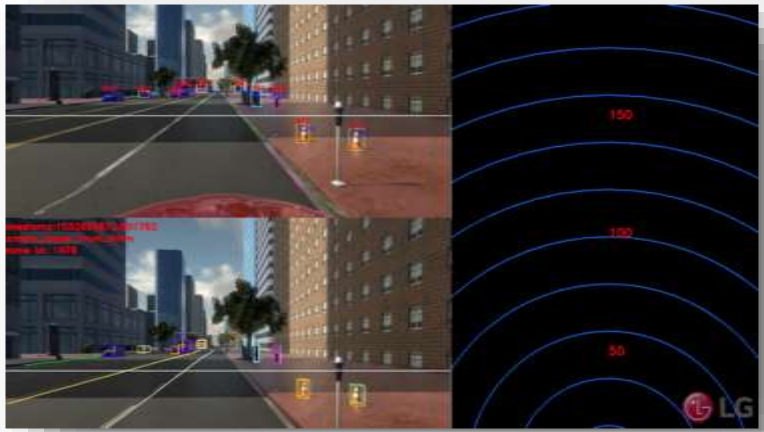
Baidu Apollo 3.0



Video

- Customized ROS
- Protobuf based bridge to the Simulator

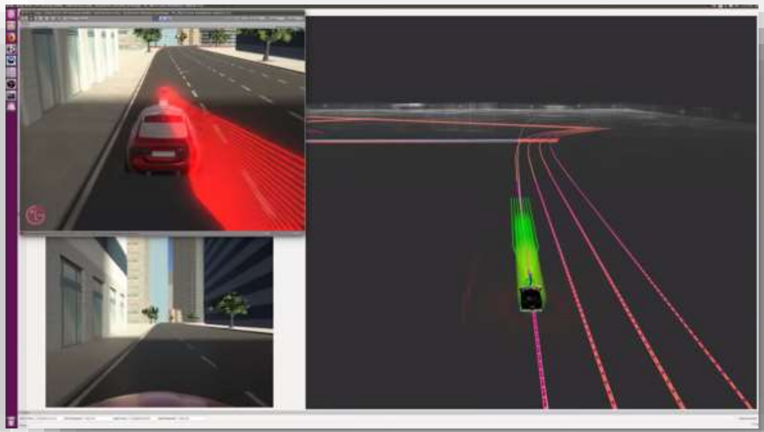
Baidu Apollo 3.5



Video

- Cyber RT
- Cyber RT based bridge to the Simulator

Autoware



Video

- ROS
- ROS Bridge to the Simulator

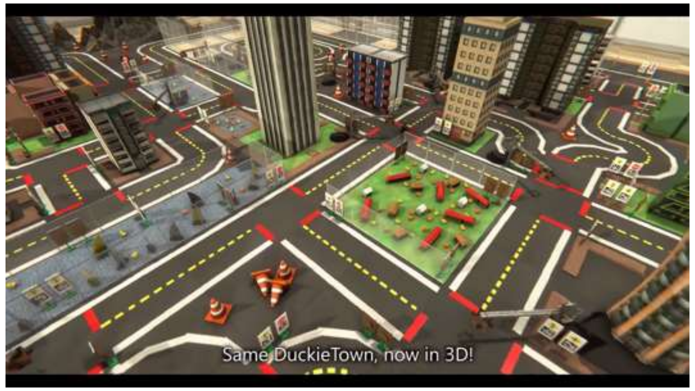
Features - Maps & Environments

Click each image to see the video

San Francisco Map **Video**

Procedurally Generated Map **Video**

Akihabara (Tokyo) Map **Video**



DuckieTown Map **Video**

Shalun Map (Taiwan Car LAB) **Video**

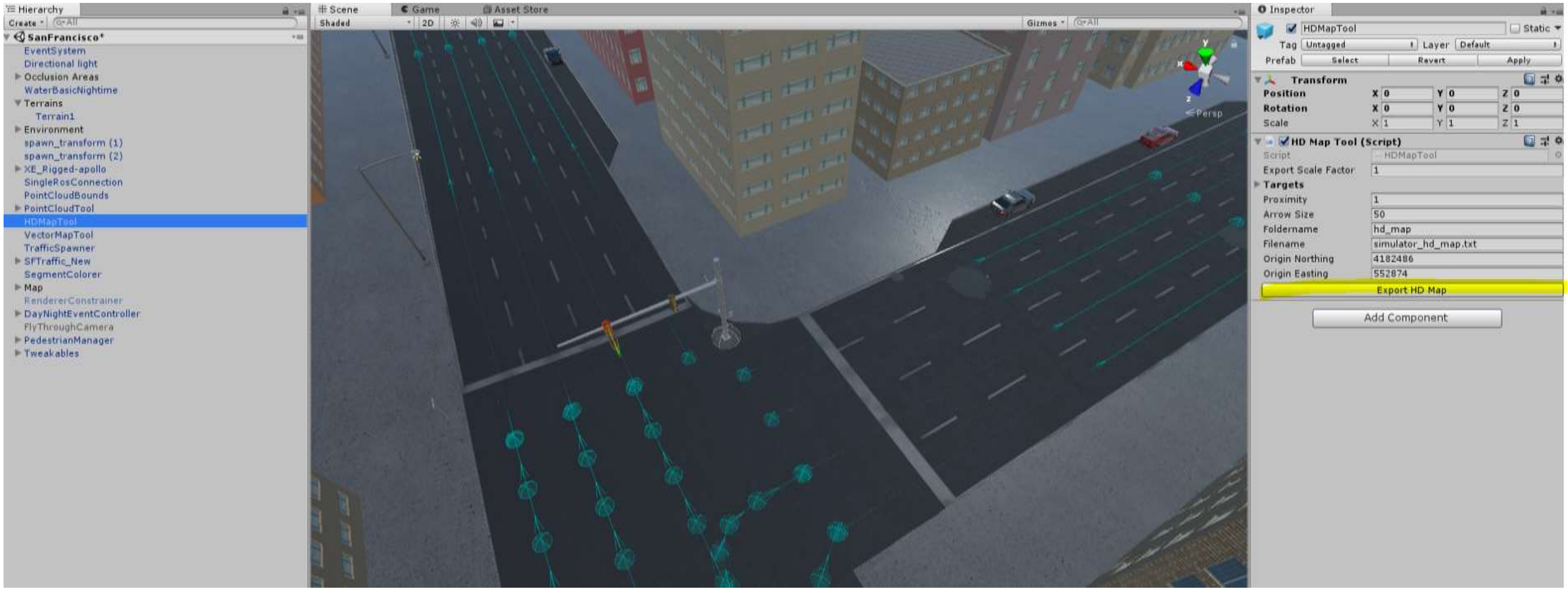
Sunnyvale (Photogrammetry & HDRP)

Features - Map Annotation Tools

YouTube Video

Supports creating, editing, and exporting of HD/vector maps of existing 3D environments (Unity scenes). The maps can be saved in the currently supported Apollo or Autoware formats.

https://www.lgsvlsimulator.com/docs/map-annotation/



SanFrancisco*
- EventSystem
- Directional light
- Occlusion Areas
- WaterBasicNightime
- Terrains
  - Terrain1
- Environment
- spawn_transform (1)
- spawn_transform (2)
- XE_Rigged-apollo
- SingleRosConnection
- PointCloudBounds
- PointCloudTool
- HDMapTool
- VectorMapTool
- TrafficSpawner
- SFTraffic_New
- SegmentColorer
- Map
- RendererConstrainer
- DayNightEventController
- FlyThroughCamera
- PedestrianManager
- Tweakables

HDMapTool — Static
Tag: Untagged — Layer: Default
Prefab: Select | Revert | Apply

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

HD Map Tool (Script)

Script	HDMapTool
Export Scale Factor	1
Targets	
Proximity	1
Arrow Size	50
Foldername	hd_map
Filename	simulator_hd_map.txt
Origin Northing	4182486
Origin Easting	552874

Export HD Map

Add Component

Features - Vehicle Models

LGSVL Simulator supports the vehicle models below. Each vehicle moves based on its own vehicle dynamics model.

Sensors for a vehicle can be configured (added, moved, etc.) in Unity Editor. (Video Link)

Jaguar XF

Video

DuckieBot (MIT)

Video

E-Palette Shuttle

Video

Tugbot (Mov.ai)

Video

Lincoln MKZ

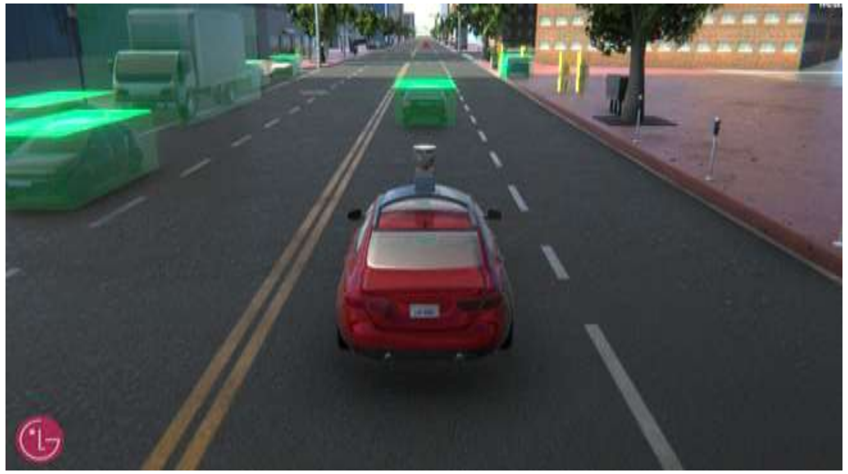
Video

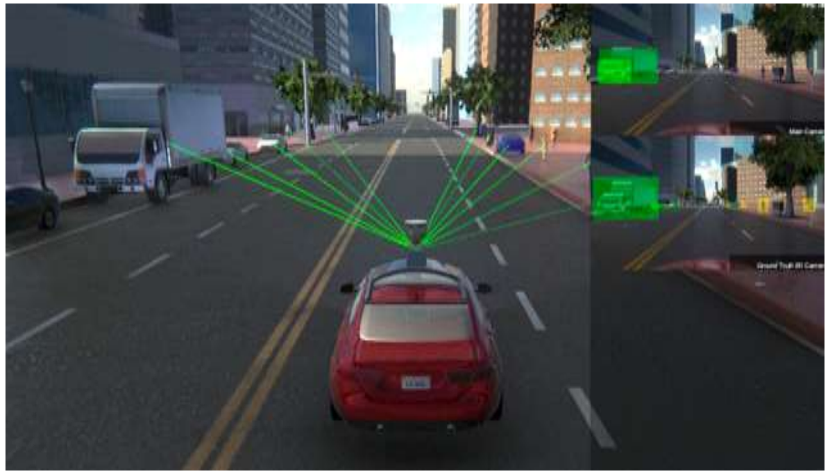
Lexus RX

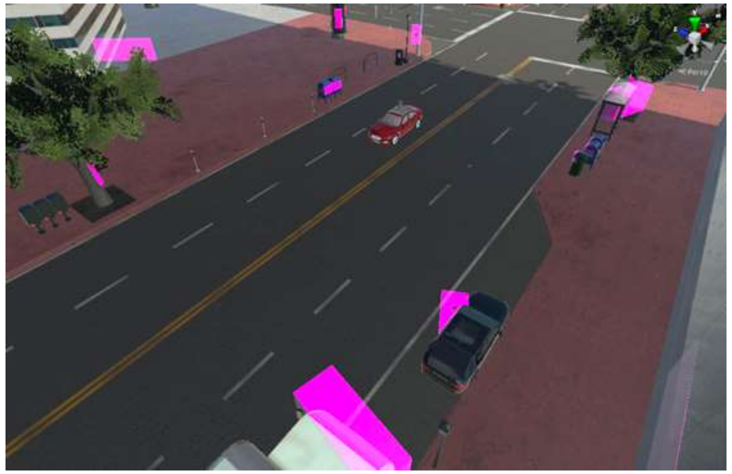
Features - Ground Truth Information & Image Segmentation

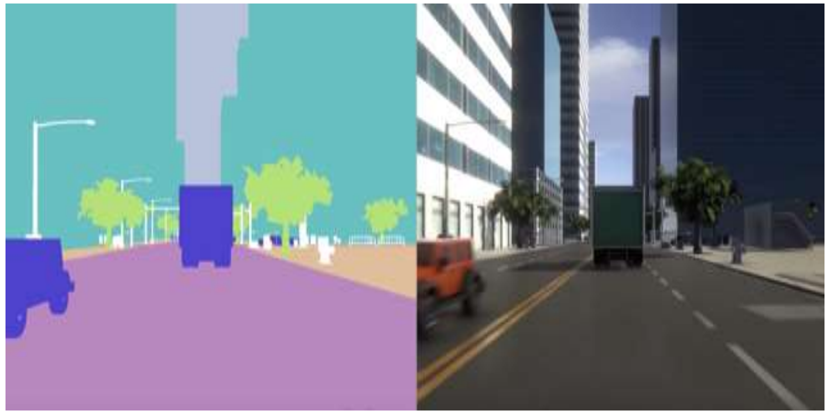
YouTube Video

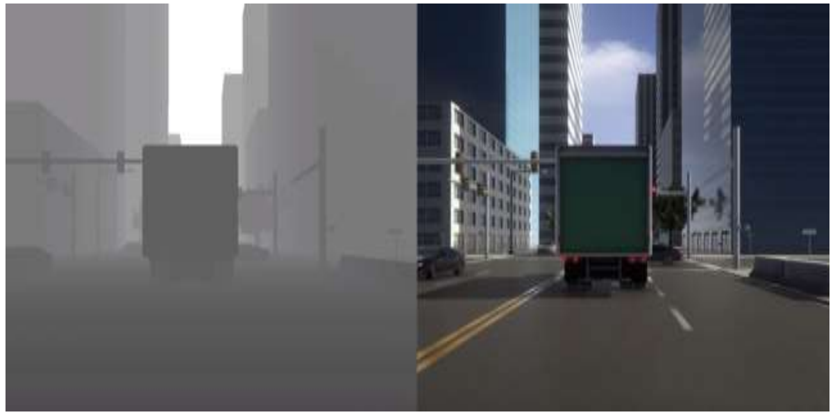
View, subscribe to, and compare ground truth obstacle information. The simulator allows visualization of 2D or 3D bounding boxes of vehicles, pedestrians, and unknown objects, and publishes detailed information (currently in a custom ROS message format) about the ground truth obstacles.







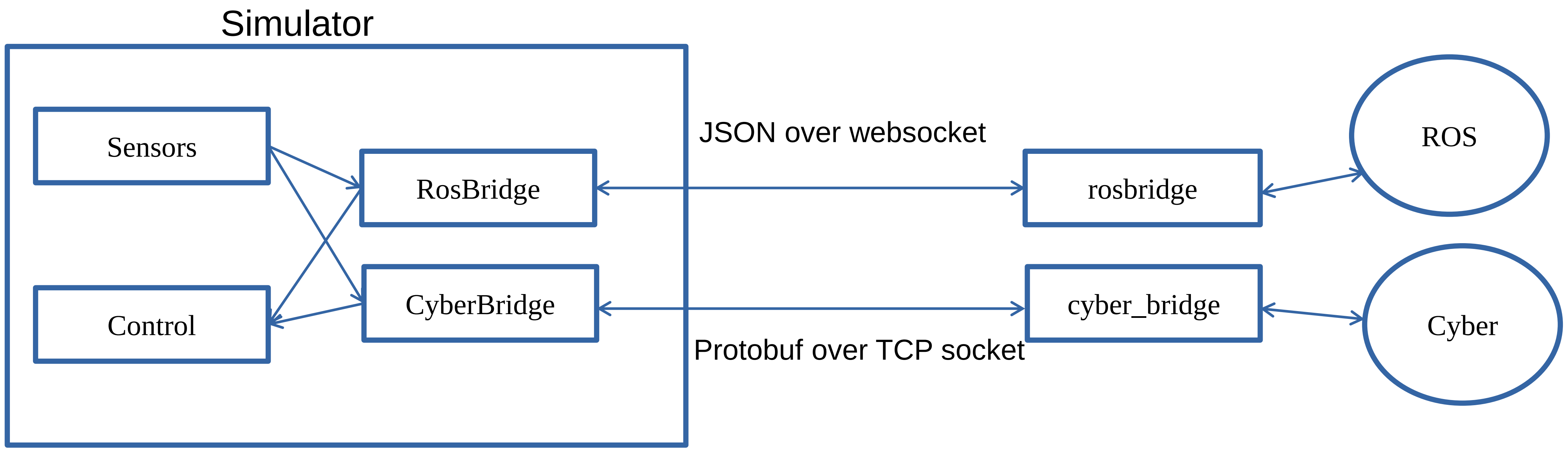




Features - AD Stack Runtime Bridge

The simulator supports ROS / ROS2, Protobuf (Apollo 3.0) and Cyber RT (Apollo 3.5).

The simulator can also support bridges for any custom runtime framework by customers.



Features - APIs

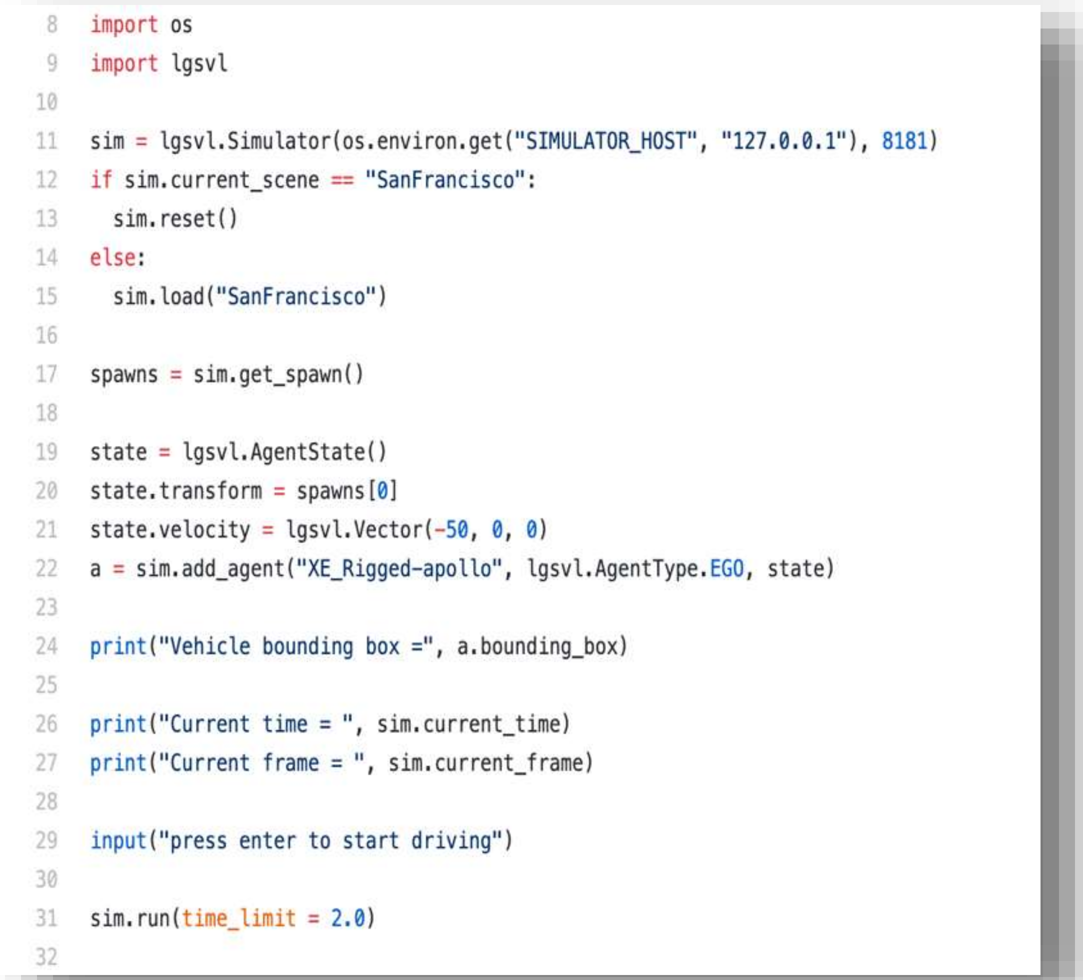
YouTube Video

Python API to configure, run, and automate LGSVL Simulator

- Create and start simulator instance
- Set up starting environment configuration
- Set up all starting agents: one or more Ego vehicles, NPC vehicles, pedestrians
- Control all vehicles (manually, waypoints, and follow HD map)
- Retrieve all sensor information
- Execute test run (duration or by time step)
- Lane change and stop line callbacks for NPCs,
- Map coordinate conversion
- And so on

https://github.com/lgsvl/simulator/tree/master/Api

https://github.com/lgsvl/simulator/tree/master/Api/quickstart



```python
import os
import lgsvl

sim = lgsvl.Simulator(os.environ.get("SIMULATOR_HOST", "127.0.0.1"), 8181)
if sim.current_scene == "SanFrancisco":
  sim.reset()
else:
  sim.load("SanFrancisco")

spawns = sim.get_spawn()

state = lgsvl.AgentState()
state.transform = spawns[0]
state.velocity = lgsvl.Vector(-50, 0, 0)
a = sim.add_agent("XE_Rigged-apollo", lgsvl.AgentType.EGO, state)

print("Vehicle bounding box =", a.bounding_box)

print("Current time = ", sim.current_time)
print("Current frame = ", sim.current_frame)

input("press enter to start driving")

sim.run(time_limit = 2.0)

```

Features - NPCs & Pedestrians

NPCs

- Each car behaves based on realistic physics PID control.
- NPCs generate a spline based on the waypoints in the HD Map and follow the Spline to smoothly travel through waypoints with little to know deviation from the centerline.

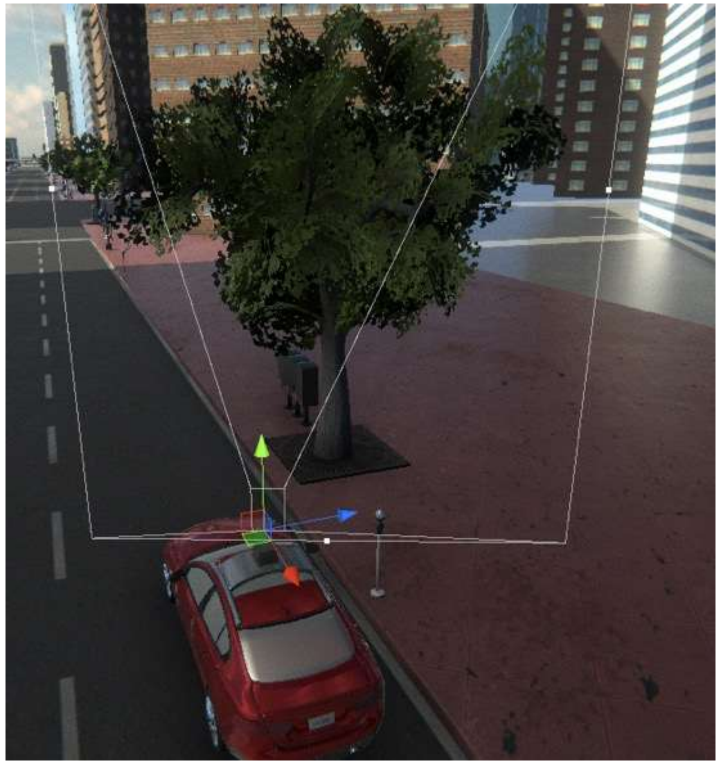
Pedestrians

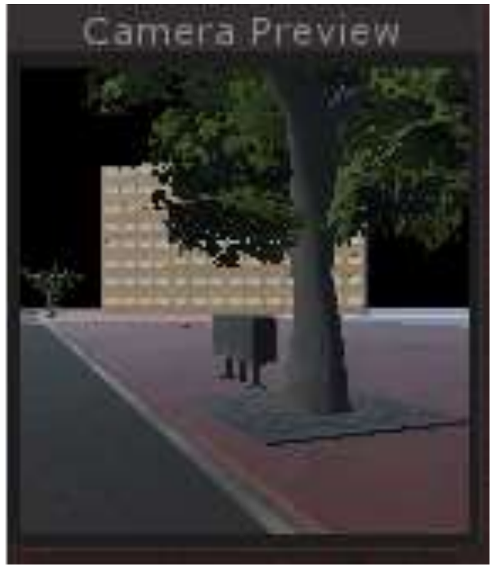
- Random pedestrians
- Random movement
- Different appearances
- Obstacle avoidance
- Random spawn locations
- Detected by all sensors: segmentation, lidar, camera

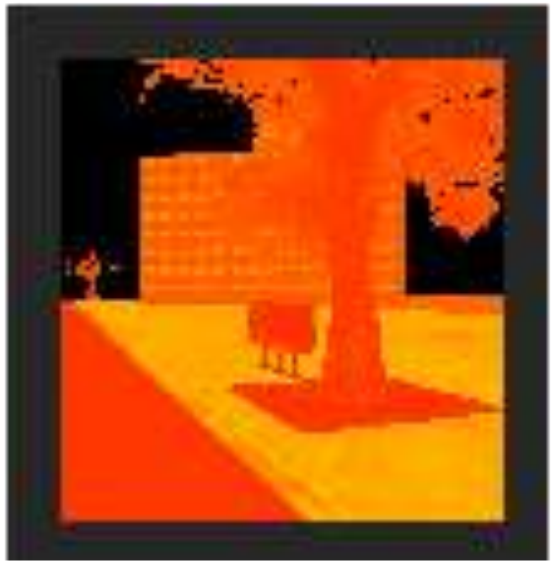
Features - GPU Accelerated LiDAR

YouTube Video

- Depth image camera: GPU retrieves depth information of all pixels in field of view, rotate field of view to simulate LiDAR
- Supported sensors: up to 128 beams





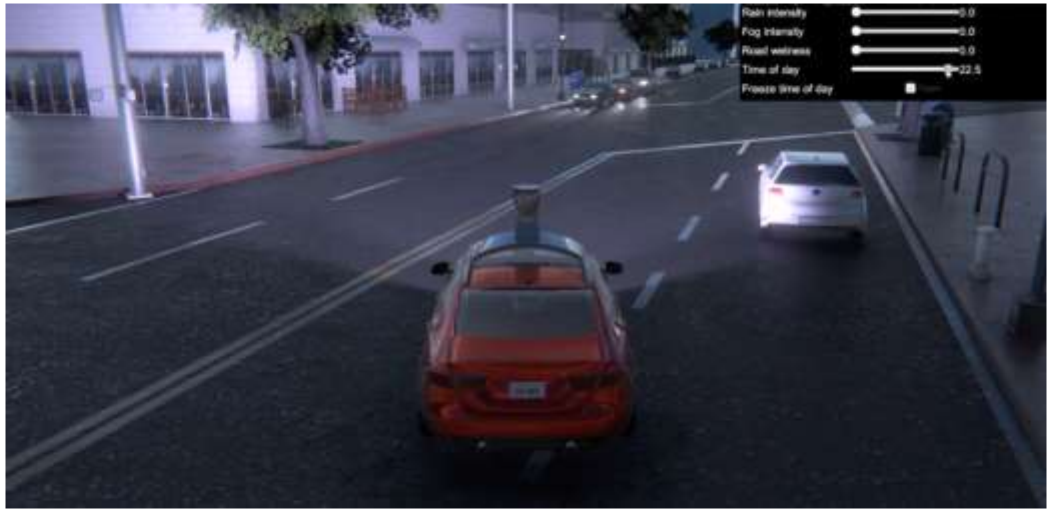


Features - Dynamic Environment

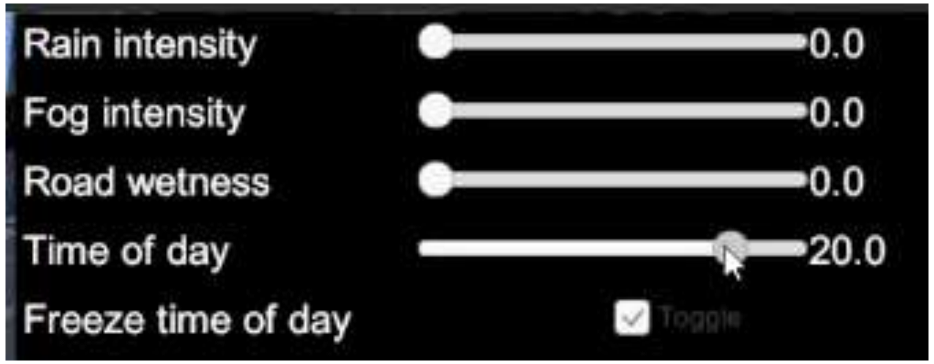
YouTube Video

- Can be controlled dynamically by Simulator UI or Python APIs

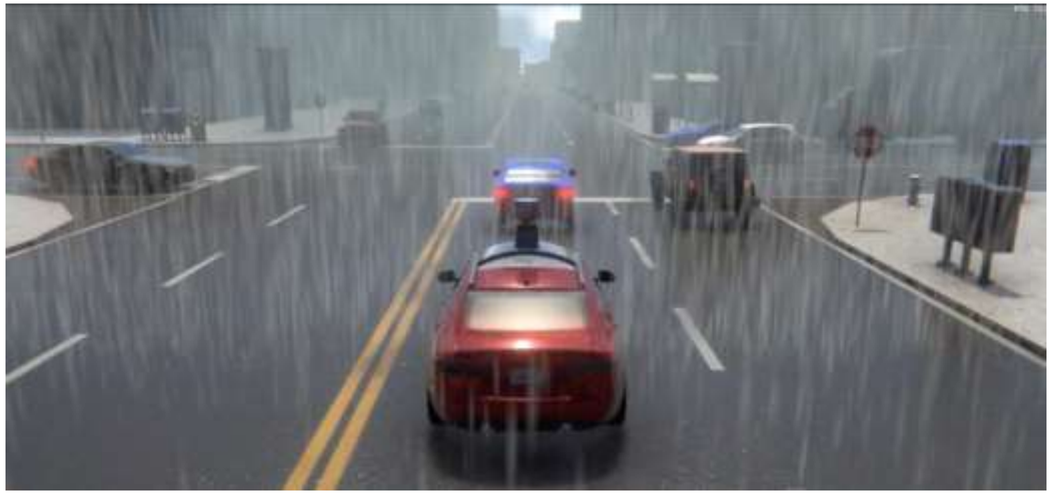
- Weather effects: rain, fog
- Time of day
- Road wetness



Rain intensity 0.0
Fog intensity 0.0
Road wetness 0.0
Time of day 22.5
Freeze time of day



Rain intensity 0.0
Fog intensity 0.0
Road wetness 0.0
Time of day 20.0
Freeze time of day ✓ Toggle

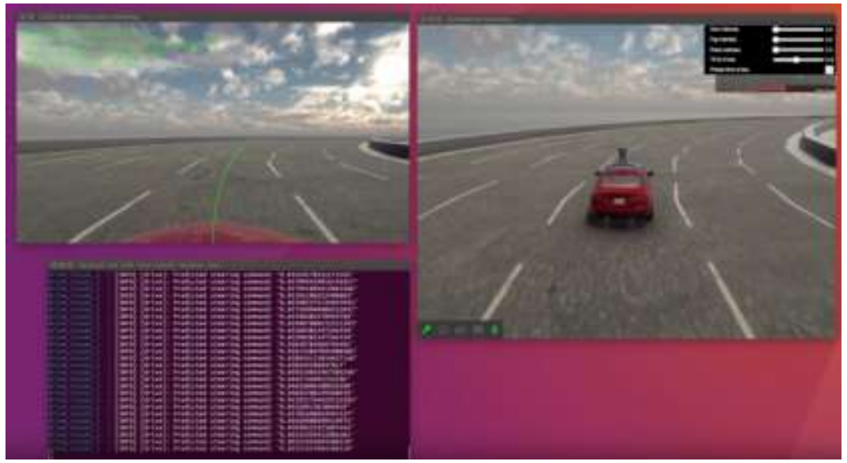


Features - Data Generation for Machine Learning

YouTube Video

Lane Following Demo based on End-To-End Deep Learning training

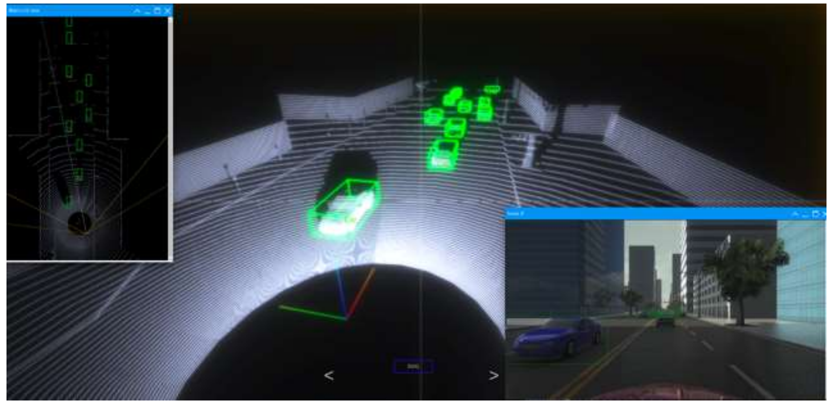
- Use LGSVL Simulator for customizing sensors (one main camera and two side cameras) for a car, collect data for training, and deploying and testing a trained model.



https://github.com/lgsvl/lanefollowing

Automate data collection and data annotation using Python API

- Collect unlimited amount of data points with Lidar point clouds, camera images, sensor calibrations and annotated them with 2D and 3D bounding boxes in KITTI format
- Train 3D object detection model with the collected data



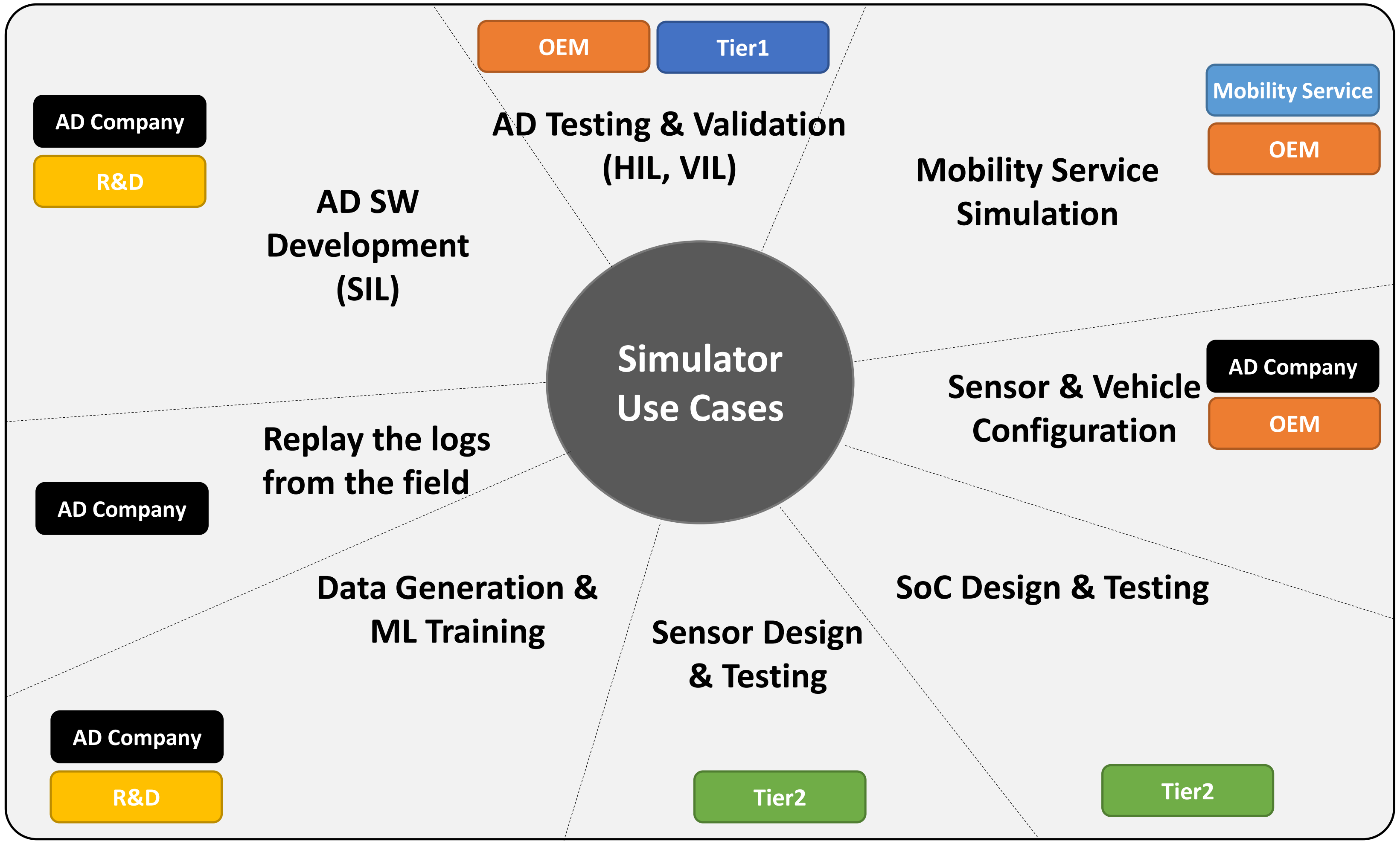
Object Detection using KITTI data

https://github.com/traveller59/second.pytorch

Simulator Use Cases

Customer Segments

AD Company	• Develop and **provide the entire or parts of AD system**
OEM	• **Validate AD system to meet safety requirements**
MaaS Company	• Mobility as a Service
Tier1	• **Integrate AD stack on HW platforms**
Tier2	• **Sensor or SoC HW Platform**
R&D	• Academia or institute **developing new algorithms or AD technologies** either in SW or HW.



Some AD companies are subsidiary of OEMs, Tier1s and MaaS companies.

Thank You !