Smart Vehicular System
Lab 1

Andrea Bedei
Lorenzo Bacchiani

University of Bologna

Index

- Open source simulator designed for research on driving scenarios
- Realistic virtual world to test vehicles with controlled weather, lighting, and traffic
- Supports training, testing, and validation while reducing costs and risks compared to on road experiments

- Safe experimentation with complex scenarios without endangering people or hardware
- Open source platform with code and protocols available on GitHub
- Open digital assets (urban layouts, buildings, vehicles) available for reuse
- Multi sensor setup with cameras, LiDAR, radar, GNSS, and other modalities for perception and planning pipelines

Why choose CARLA

- Widely adopted by industry and academia with many published baselines and example projects
- Active community with extensive documentation and shared troubleshooting resources
- Native ROS 2 interface and ROS bridge for ROS 1 and ROS 2 integration

CARLA requirements (summary)

Area	Requirement
Supported systems	Windows or Linux (64 bit)
GPU	Dedicated GPU with at least 6 GB VRAM, 8 GB recommended
Disk space	About 20 GB for the packaged simulator
Python	Python 3 with pip 20.3+; pygame and numpy for quick start
Network	Two TCP ports available, 2000 and 2001 by default

Note: CARLA can also run with a discrete GPU (not integrated) with at least 4 GB VRAM and sufficient system RAM (16 GB+).

Operational constraints

- Setup can be demanding and benefits from familiarity with Python and simulator configuration
- Simulation is an approximation of reality, so traffic and pedestrian behavior can differ from real urban dynamics
- Official support focuses on Linux and Windows, with limited support for other platforms and architectures

Get the simulator binaries

If you want to use your own PC, download CARLA 0.9.15 from the official site:

- GitHub releases: `https://github.com/carla-simulator/carla/releases`

If you use the lab PCs, CARLA is already available. Follow these steps:

- Open File Explorer
- Go to This PC
- Open Local Disk (C:)
- Enter the `Carla_0.9.15` folder

Run the simulator

- After extracting the package, start CARLA via the graphical launcher or the server executable
- Typical workflow: download CARLA, set up the Python client, then run the example scripts once the server is running
- Keep the default networking ports available and ensure the firewall does not block the connection

Python environment setup

Use a Python environment manager (conda or venv) to keep dependencies isolated.

Create a conda environment

```
conda create -n carla-env python=3.7
conda activate carla-env
python --version # Python 3.7.x
```

Install dependencies and the CARLA Python package

```
pip install -r PythonAPI/examples/requirements.txt
pip install carla
```

Prerequisite: CARLA server must be running.

```
cd Code/examples
python tutorial.py
```

- Connects to the server and spawns a vehicle
- Enables autopilot and attaches a depth camera
- Saves images in _out/

Exercise 2: manual_control.py

Prerequisite: CARLA server must be running.

```
cd Code/examples
python manual_control.py
```

- Keyboard driving with HUD (demo live)
- Main keys: WASD, P (autopilot), H (help)

Prerequisite: CARLA server must be running.

```
cd Code/examples
python dynamic_weather.py
```

- Changes sun position and weather over time
- Clear visual feedback for lighting and rain

Prerequisite: CARLA server must be running.

```
cd Code/examples
python generate_traffic.py
```

- Spawns vehicles and pedestrians automatically
- Options: `--number-of-vehicles` and `--number-of-walkers`

Python API quick connection

```python
import carla

# Connect to the server
client = carla.Client('localhost', 2000)
world = client.get_world()
```

1. A Dosovitskiy, G Ros, F Codevilla, A Lopez, V Koltun. CARLA: An Open Urban Driving Simulator. Proceedings of the First Annual Conference on Robot Learning, 2017.

2. CARLA documentation, version 0.9.15. `https://carla.readthedocs.io/en/0.9.15/`

3. CARLA GitHub repository. `https://github.com/carla-simulator/carla`

4. ROS website. `https://www.ros.org`

Thank you for your attention

Andrea Bedei

Lorenzo Bacchiani

University of Bologna