

## LAB 1

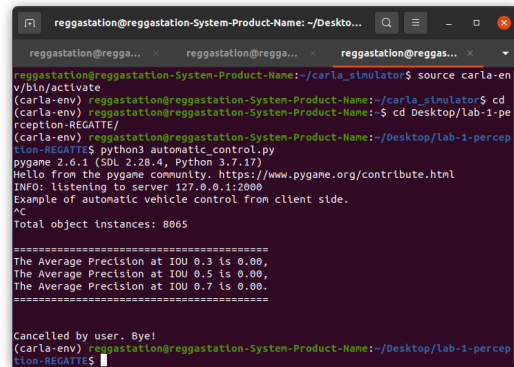
## 1 Part 1: Environment Setup

**Platform:** I built the caria simulator locally on my personal PC. Created a virtual environment that supports both PyTorch and Tensorflow.

**Results:** After running the `automatic_control.py` from the repositories starter code, I get the following output:



(a) PyGame Window output



(b) Terminal Output

Figure 1: Part 1: Setting up Environment

## 1.1 To-Do

1. Activate the virtual environment and launch Carla
2. **Automatic Control:** cd to the GitHub repo, and run `python3 automatic_control.py`. The car drives automatically, but crashes into other cars or hits pedestrians or breaks traffic rules.
3. Once you end the simulation by pressing `Ctrl + C`, the 1(b) output is received.

## 2 Part 2: Setup new sensors and visualize ground truth

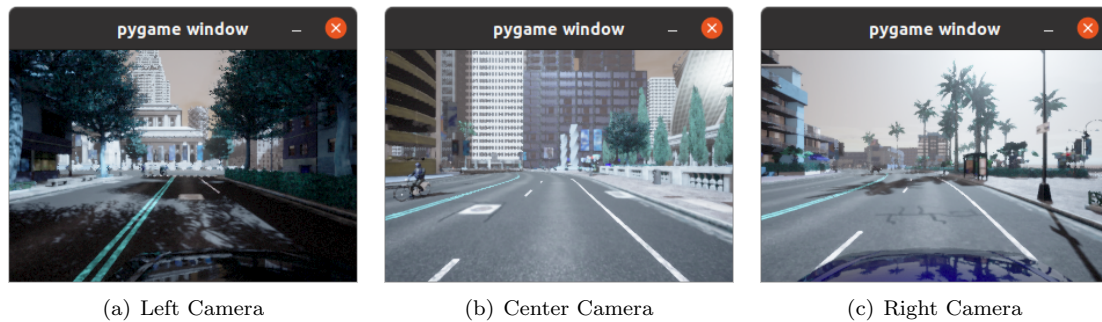


Figure 2: Camera output from custom placements

### 2.1 Sensor Placement & Reasoning

I used 3 RGBA cameras for this lab where I wanted to setup a camera system to work as follows:

1. **Center Camera:** Main camera to run Object Detection
2. **Left & Right Camera:** Use 2 camera inputs, to estimate depth.

#### 2.1.1 Sensor Values

```
sensors = [  
    {'type': 'sensor.camera.rgb', 'x': 1.0, 'y': -0.4, 'z': 1.60, 'roll': 0.0, 'pitch': 0.0, 'yaw': 0.0,  
     'width': 300, 'height': 200, 'fov': 100, 'id': 'Left'},  
  
    {'type': 'sensor.camera.rgb', 'x': 1.0, 'y': 0.4, 'z': 1.60, 'roll': 0.0, 'pitch': 0.0, 'yaw': 0.0,  
     'width': 300, 'height': 200, 'fov': 100, 'id': 'Right'},  
  
    {'type': 'sensor.camera.rgb', 'x': 2.0, 'y': 0.0, 'z': 1.60, 'roll': 0.0, 'pitch': 0.0, 'yaw': 0.0,  
     'width': 300, 'height': 200, 'fov': 100, 'id': 'Center'},  
  
    {'type': 'sensor.other.gnss', 'x': 0.7, 'y': -0.4, 'z': 1.60, 'id': 'GPS'}  
]  
return sensors
```

Figure 3: Sensor Values for 2(a), 2(b), 2(c)