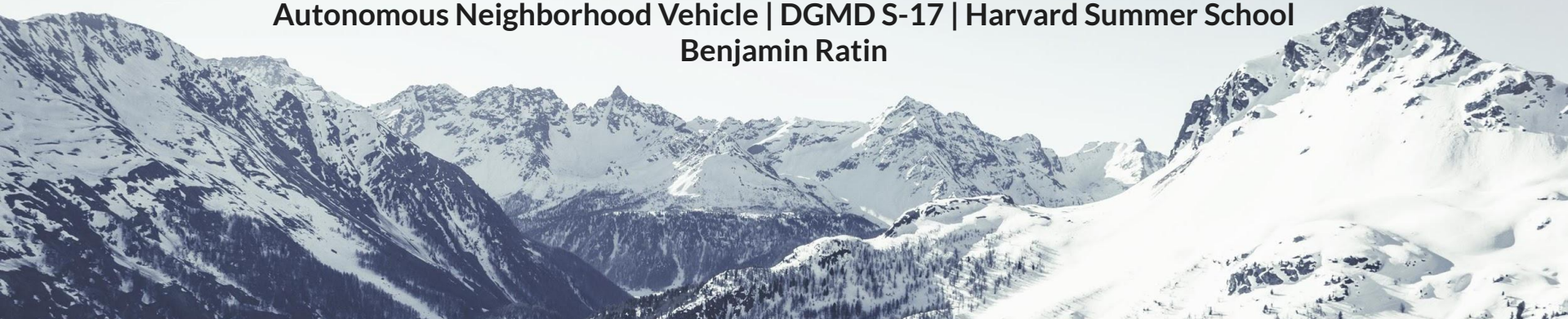
Think Green
Think Clean



Autonomous Neighborhood Vehicle | DGMD S-17 | Harvard Summer School

Benjamin Ratin

Introduction

Team

Benjamin Ratin

I am a rising senior high school student in Newton, MA.

I am working on the project on my own.

Goal of the Project

To build a crawler vehicle that can drive autonomously and cover every street in a neighborhood to perform a useful task.

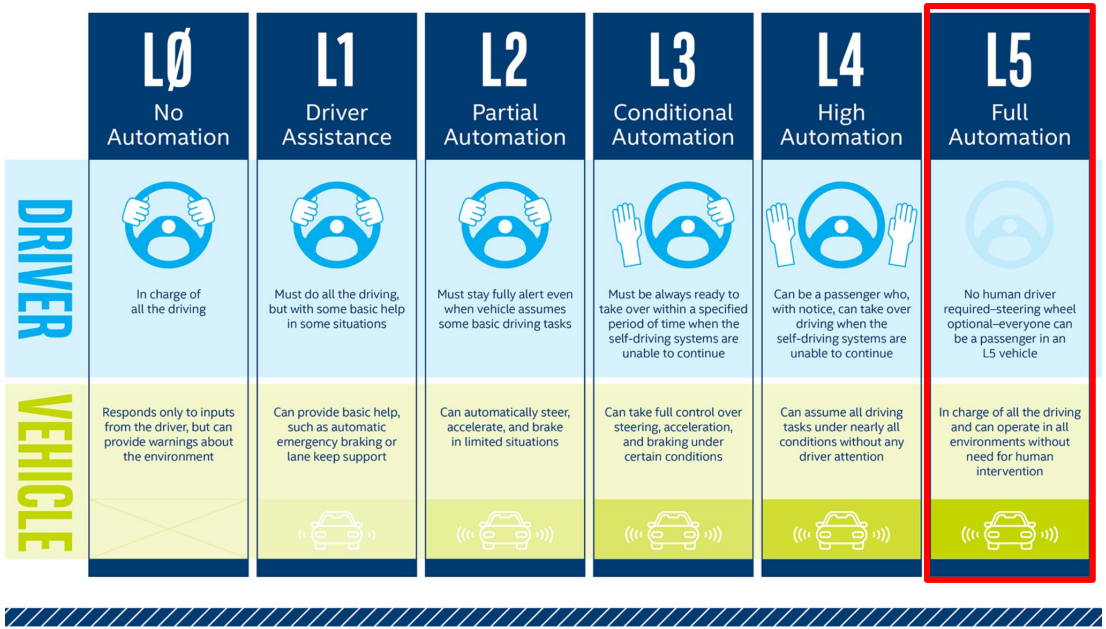
Problems to solve

1. Street cleaning
2. Garbage collection
3. Checking for safety hazards
4. Parking enforcement

Level of Autonomy

Level 5

Full Automation



Sources: Society of Automotive Engineers (SAE); National Highway and Traffic Safety Administration (NHTSA).

Related Work

- Materials from this course
- Inspired by the “Interceptor 4” project presented in the first lecture.
- Carla simulator, chosen after looking at multiple options
- Carla’s Python API, that I am using extensively in my code
- Multiple other Python libraries, most notably OpenCV and NumPy
- ROS architecture and tutorials (although I did not use ROS directly for this project)
- Basic prior knowledge of graph theory

Technology

Software

- The logos on the right represent software that I used
- Carla documentation
- OpenCV documentation
- A number of helpful links and tutorials are referenced in the code and the slides above
- Some OpenCV functions were originally developed on Ubuntu 20.04

Hardware

- Windows desktop, Intel Core i7, 32 GB RAM, NVidia GTX 970 GPU
- No robotics hardware - I am using a simulator

Choice of a Simulator

Carla Simulator (https://carla.org/) chosen:

- Based on robust Unreal engine
- Built-in realistic town maps
- Easy to use Python API with plenty of examples
- High quality documentation
- Existing libraries to assist with autonomous vehicle development (such is low-level driving controls)

Other options considered:

- Unity
- Gazebo

Features

Description

The inputs are:

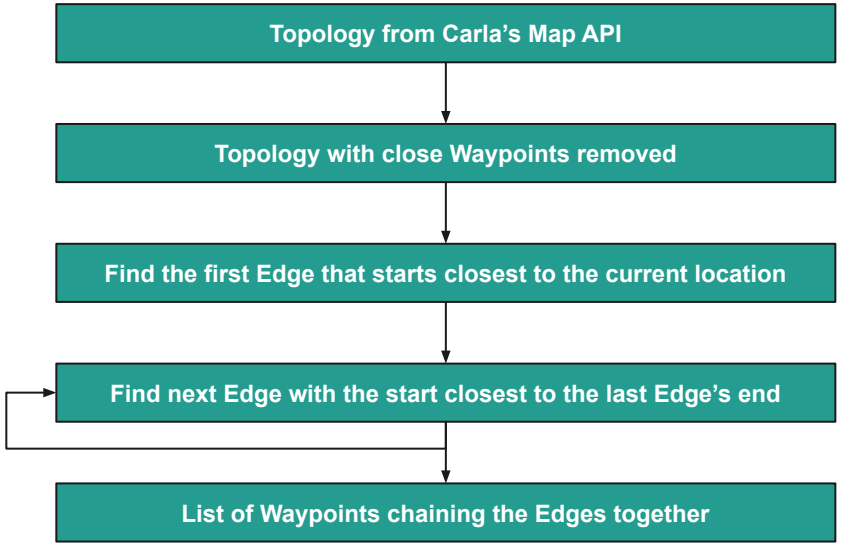
- A map of the area
- A task to perform (out of scope of the project)

The vehicle drives over every street on each side based on the map. It obeys traffic laws, although this is largely handled by the simulator libraries.

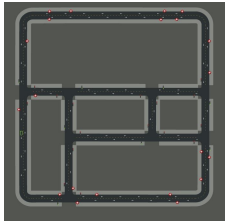
Addressed challenges (for the rest I relied on Carla):

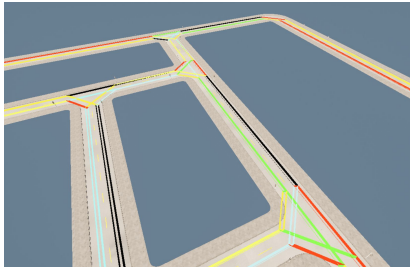
- Route planning to cover all streets and intersections in both directions
- Real time visualization of the route in progress
- Object detection to help perform the task

Route Planning



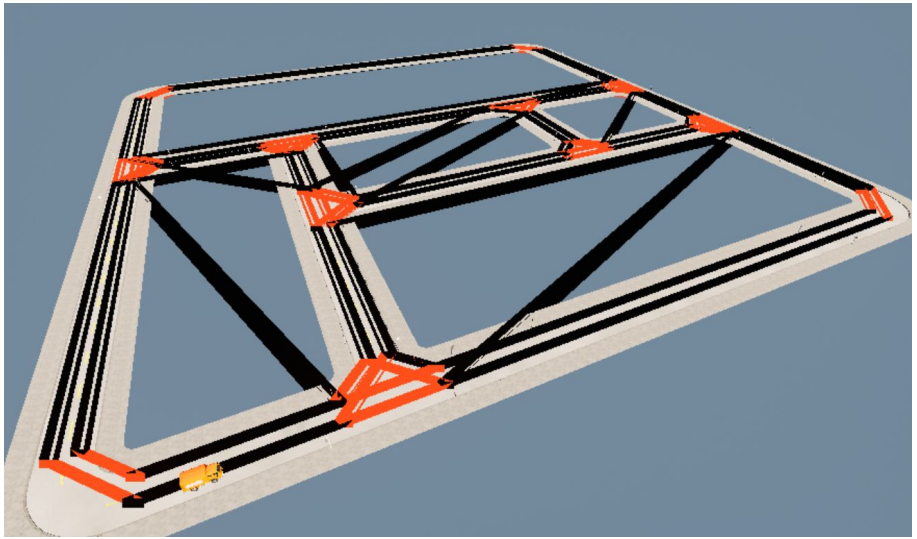
Town02 map from the Carla simulator



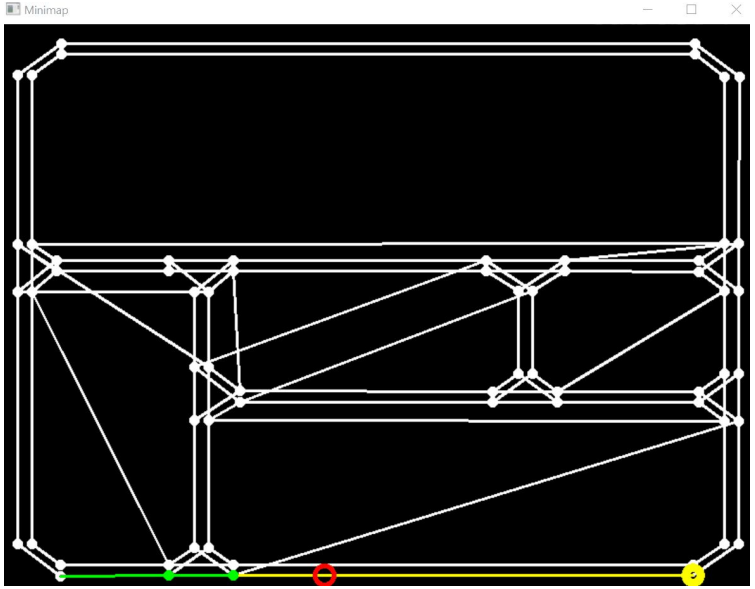


Town02 map from the Carla simulator

Complete Plan



Complete plan with everything covered



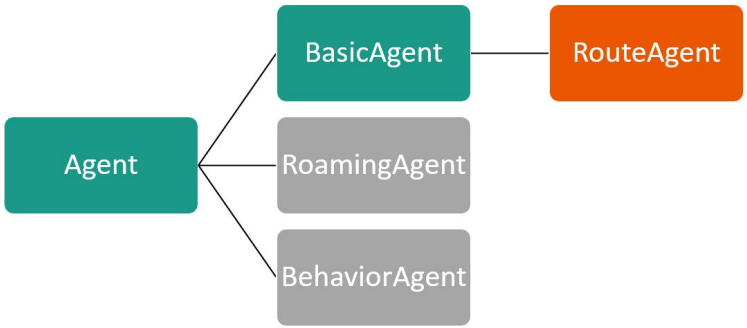
Autonomous Driving

- Low-level driving is done by the Carla simulator
- I control the destination list based on the computed route
- I experimented with lane detection and object detection

Agent - implements common functions for all agents

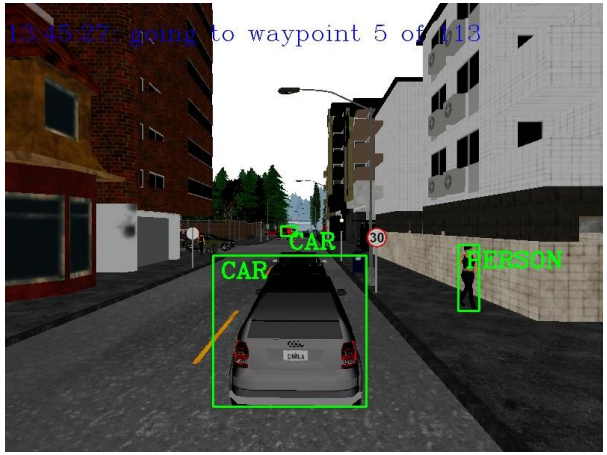
BasicAgent - navigate scene to reach a specified destination

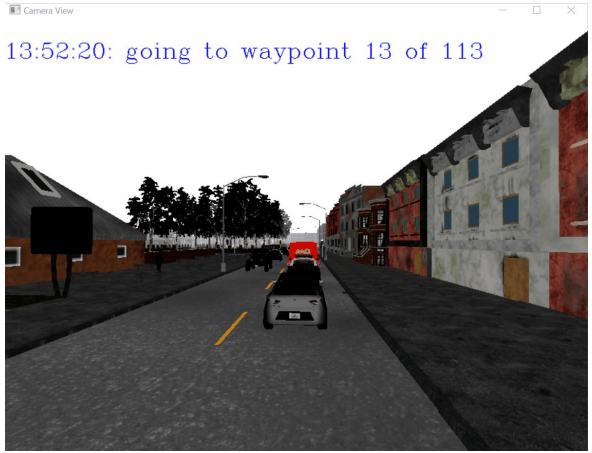
I implemented a ***RouteAgent*** class that can maintain a list of routes and guide its parent class to drive to each destination in sequence.



Object Detection

- Object detection is required for
  - driving
  - performing the vehicle tasks
- I am using **dnn_DetectionModel** method available in OpenCV with a pre-trained model
- Allows for detection of basic objects





References: https://www.youtube.com/watch?v=HXDD7-EnGBY , https://github.com/ankityddv/ObjectDetector-OpenCV.git

Demo

Show Demo

Conclusion

What I Learned From This Project

- ROS and Carla architecture
- Carla API
- OpenCV and NumPy API
- Object detection
- Graph theory algorithms
- Linear algebra

Future Work

- Include a task execution, like a garbage bin, parked car, etc.
- Improvements to the route selection algorithm
- Better training on the object recognition model (on Carla objects)
- Object tracking
- Real AI driving based on sensors
- Use of ROS bridge and open source ROS Nodes

Thank you.

Project Schedule

Milestones	Date(s)	Comments
Familiarize myself with robotics development tools, like ROS	6/22 - 7/10	
Pick a simulation engine	7/3-7/13	Looked at Unity, Gazebo and a few other options
Setup development environments	7/3-7/14	Setup a Windows and an Ubuntu dev environments with all the tools ready
Basic self-driving car using existing Carla tools	7/16	Have basics working and going through more complex tutorials
Routing/navigation module prototype	7/25	I will start with a standalone set of functions that can plot the route among waypoints
Routing/navigation module fully integrated	7-29-8/1	
Identification of possible improvements to the overall system	Ongoing	
Work on the presentation with demo video(s)	7/28-8/3	
Final presentation	8/5	