Table of Contents

Features Implemented 2

Map Destination Feature 2

Autonomous Vehicle Navigation 2

Data Saver 2

Libraries Used 2

Test DataSet 3

Running Instructions 3

Screenshots 4

Main Menu Screenshot 4

Set Map Destination 4

Driving Screenshot 7

Diagnostics Screenshot 8

References: 10

Autonomous Vehicle Implementation Details

The autonomous vehicle has been implemented by following the design document in part 1. Due to the complexity of the autonomous vehicle, not all features could be implemented.

Features Implemented

The following features are working.

Map Destination Feature

Allows the user to specify the source and destination addresses for moving the autonomous vehicle.

Autonomous Vehicle Navigation

The navigation feature has been implemented; however only steering wheel has used to control the autonomous vehicle. The navigation feature was implemented using a neural network which had to be trained using a publicly available dataset namely (Boeing, 2017; Fouad et al., 2019).

Data Saver

A data saver module has been implemented which saves the diagnostic data. The data which has been saved includes the following attributes:

- Source and Destination Address
- Map Route attributes like street name, highway information and max speed of the road
- Steering angle of the autonomous vehicle when its moving

Libraries Used

The following libraries have been used for implementing the design.

1) **Folium:** For displaying OpenStreetMap of the route

2) **OMNx:** For interacting with OpenStreetMap street network and getting the shortest route from source address to the destination address. It has also been used to convert the address into the Latitude, Longitude format which is required for getting the route.

3) **Pickle module**: For saving the route, map and driving parameters to the disk and retrieving back. This is used by the Data Saver module.

4) **OpenCv:** For driving Vehicle and rotating the steering wheel (Boeing, 2017; Haklay and Weber, 2008).

Test DataSet

Three rows have been used to test the functionality of the street network and routing from the OSMNx API. Out of three tests performed, two tests have passed while the third test indicated in red has failed. The failure for the third test is being investigated (Boeing, 2017).

Source	Destination	Route & Speed limit	Actual
London Eye	London Borough	Tower Bridge Roa, 30mph	Tower Bridge Road, 30mph
London Eye	Reading	Stevens Street, 20mph	Stevens Street, 30mph
Camden Town	Reading	40mph	30mph

Python files, driving dataset modules instructions, test dataset.

Running Instructions

All the modules features, and functionality is accessible from the main GUI module named as frontend.py

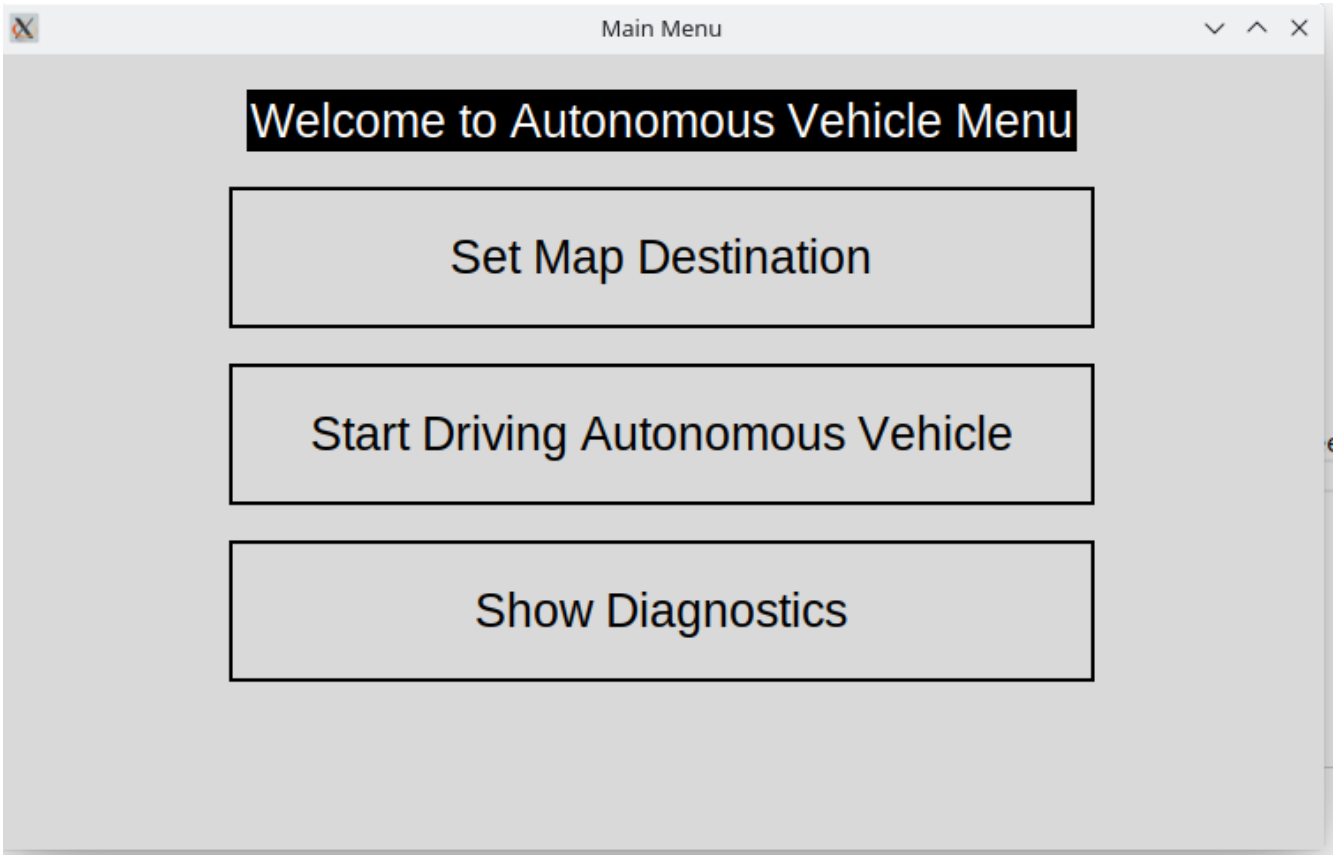
To run it:

python3 FrontEnd.py

Pip Package Requirements: Folium, OSMNx, Pickly

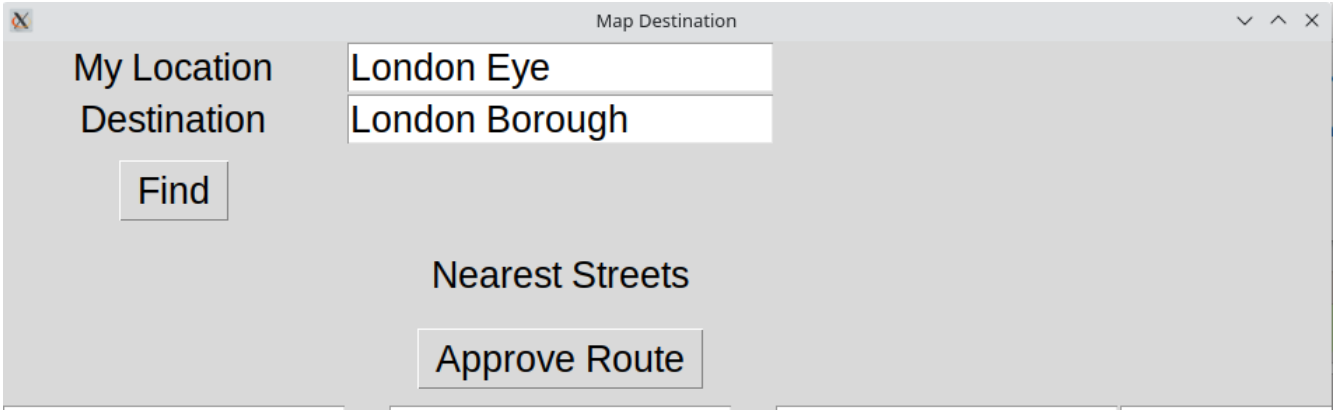
Screenshots

Main Menu Screenshot

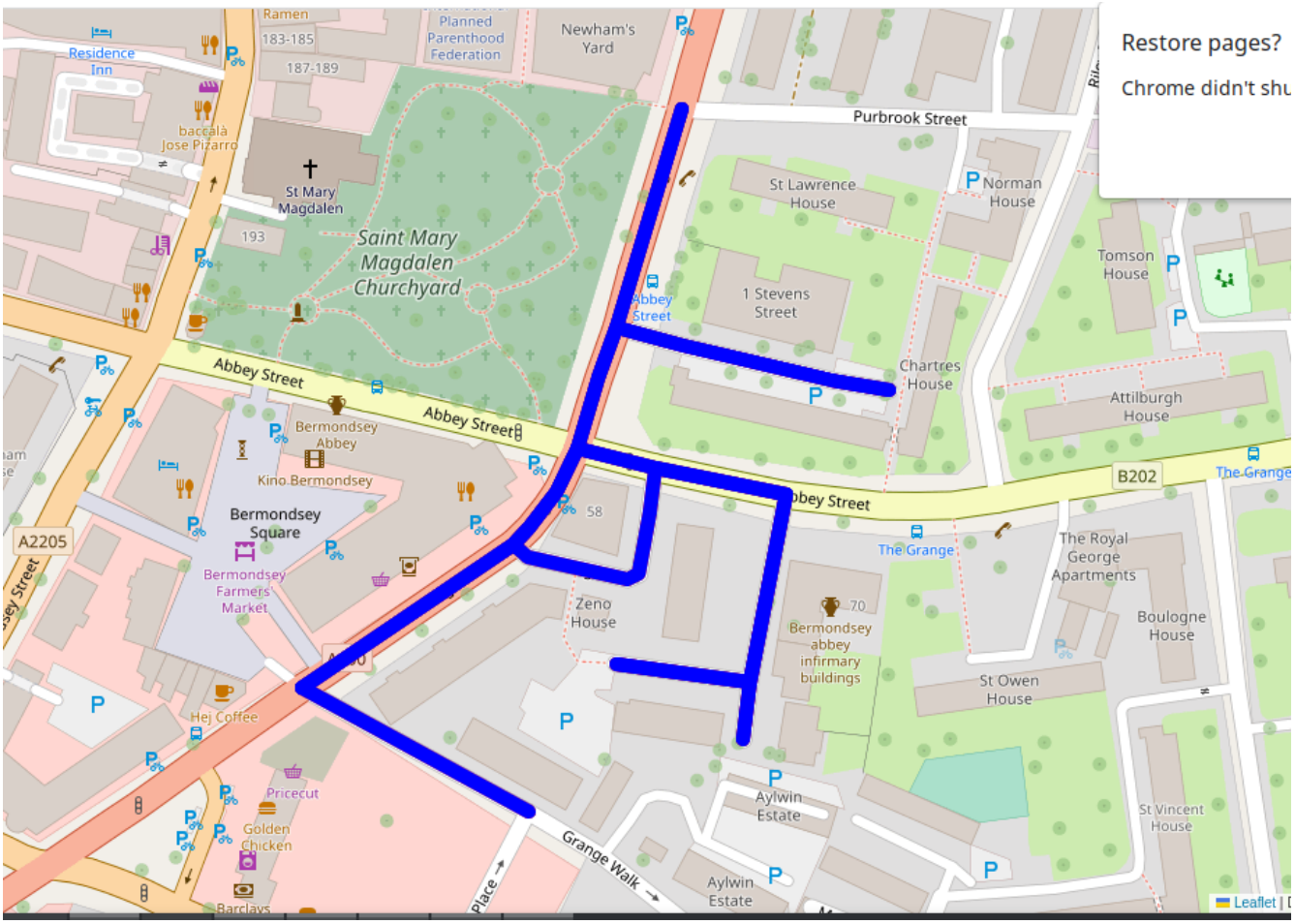


Set Map Destination

Displayed when user clicks on Set Map Destination. User has to enter source address and the destination address and then click Find. The system would automatically find nearby streets and routes for moving from source to destination.

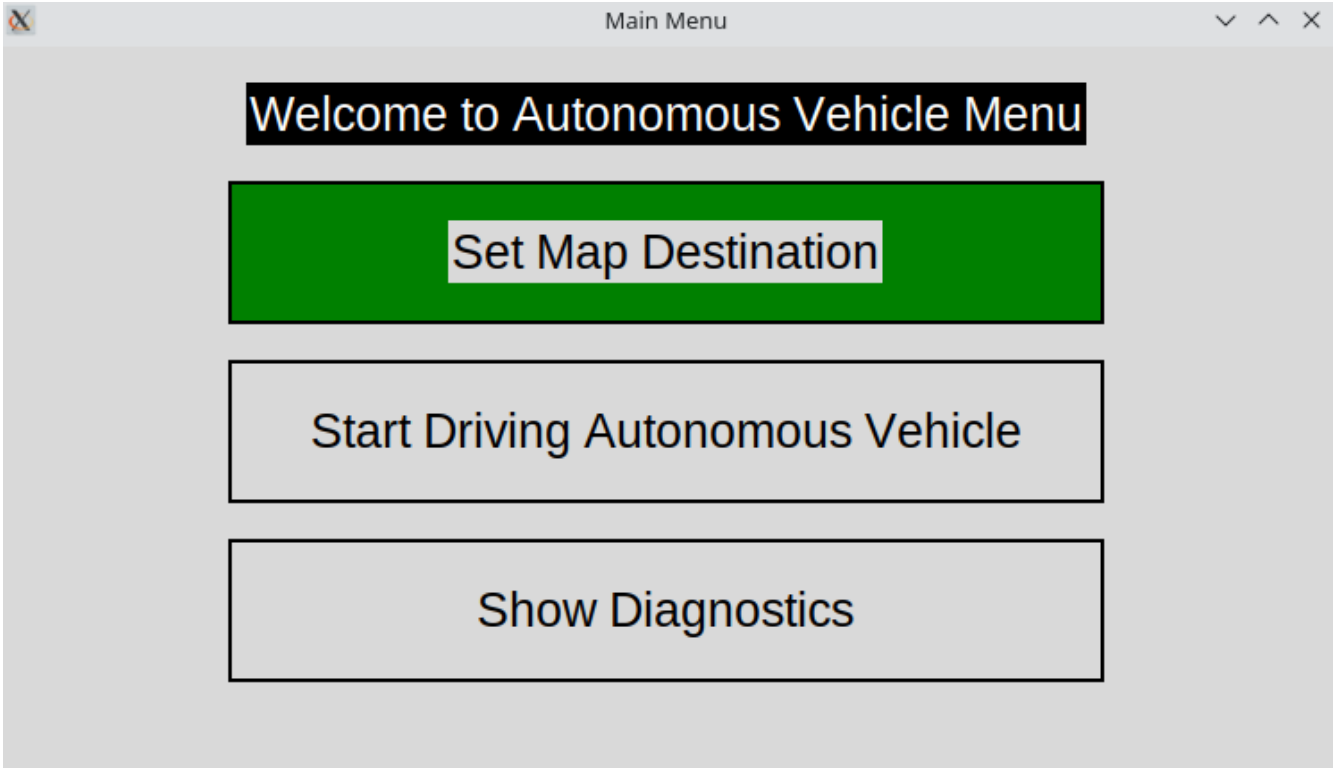


OsmID	Name	Highway	MaxSpeed
655513313	Tower Bridge Road	trunk	30 mph
655513314	Abbey Street	secondary	20 mph
1069976326 655513311	Tower Bridge Road	trunk	30 mph
26231686	Grange Walk	residential	20 mph
655513312	Tower Bridge Road	trunk	30 mph
25960266 59273212	Long Walk	residential	20 mph
1069976326 655513311	Tower Bridge Road	trunk	30 mph
655513312	Tower Bridge Road	trunk	30 mph
670612944 1069976325	Tower Bridge Road	trunk	30 mph
666136828	Abbey Street	secondary	20 mph



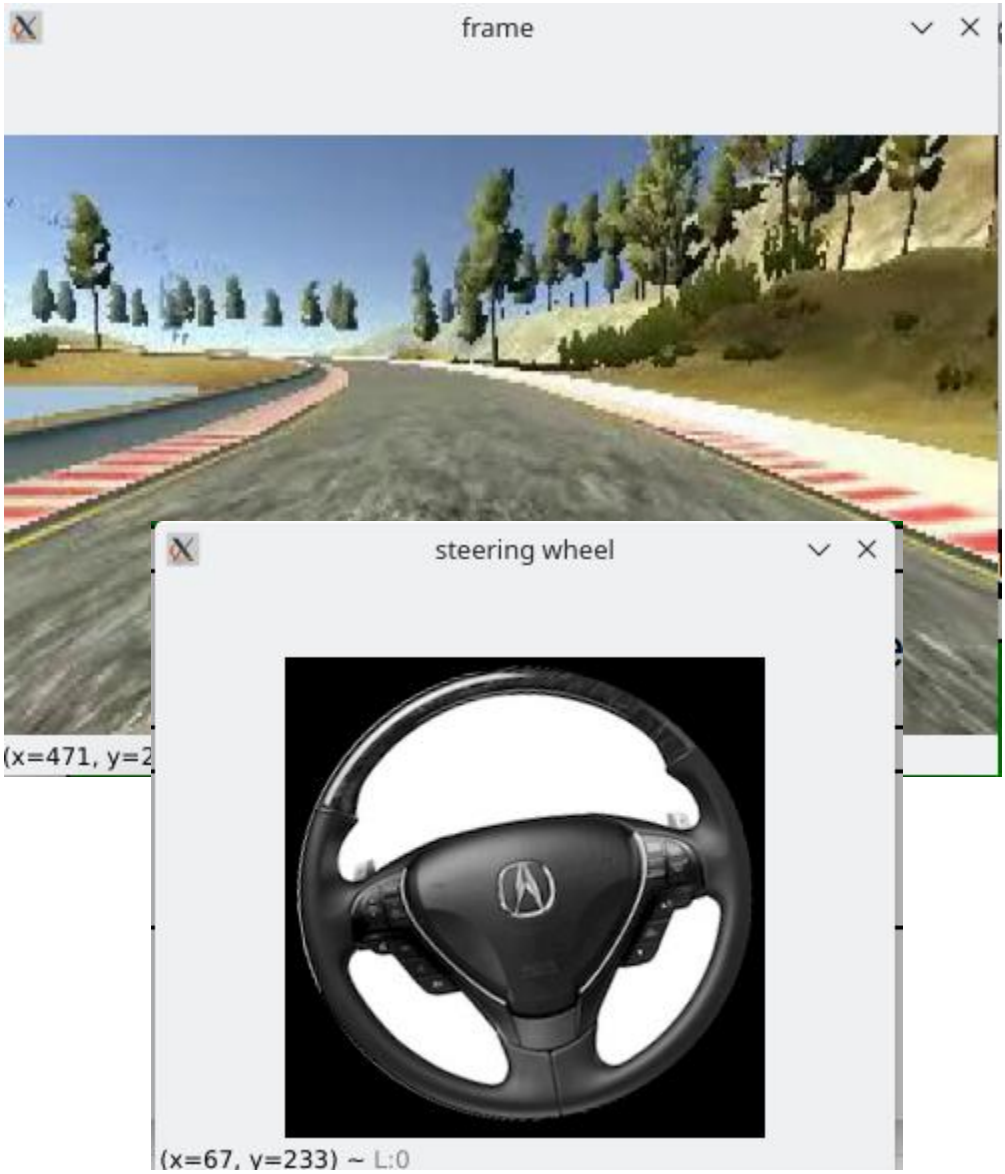
The user then pressed the approve route button to set the destination of the autonomous vehicle. Approve routes calls the findShortestRoute method of the map class that finds the shortest route between source and destination.

After this, Set Map Destination is turned green, showing this operation has been completed as given below



Driving Screenshot

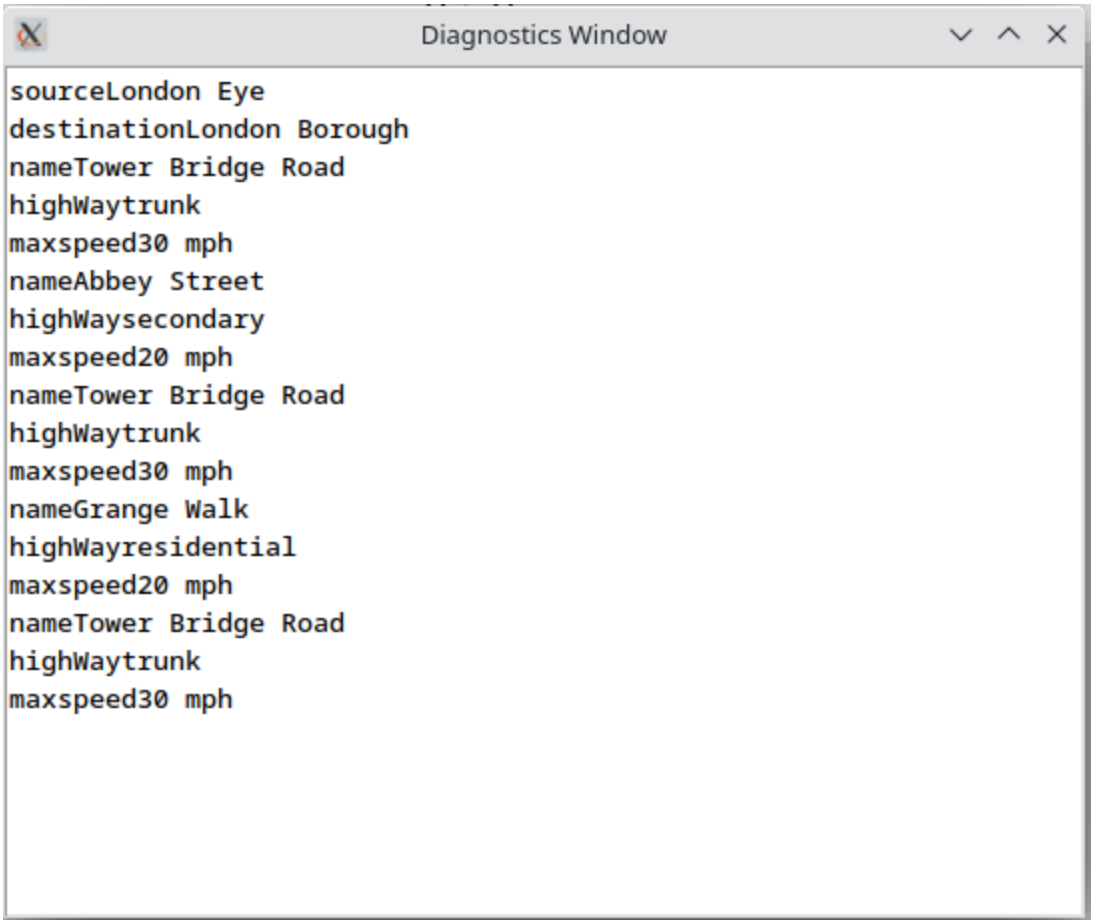
Uses a machine learning model for training the autonomous vehicle on the Udacity Dataset based on Udacity Simulation (Udacity, 2016).



The steering wheel rotation comes from Keras model, which has been trained. Takes input the image of the scene and outputs a steering angle using keras predict function.

Diagnostics Screenshot

In the main menu, when the user clicks on the “Show Diagnostics” the diagnostic data of the autonomous vehicle is shown.


Diagnostics Window
sourceLondon Eye
destinationLondon Borough
nameTower Bridge Road
highWaytrunk
maxspeed30 mph
nameAbbey Street
highWaysecondary
maxspeed20 mph
nameTower Bridge Road
highWaytrunk
maxspeed30 mph
nameGrange Walk
highWayresidential
maxspeed20 mph
nameTower Bridge Road
highWaytrunk
maxspeed30 mph

References:

1. Boeing, G., 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, *65*, pp.126-139.
2. Haklay, M. and Weber, P., 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, *7*(4), pp.12-18.
3. Fouad, A.M., Sharkawy, R.M. and Onsy, A., 2019, October. Fixed obstacle detection for autonomous vehicle. In *2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE)* (pp. 217-221). IEEE.
4. *2016. Udacity self driving car. https://github.com/udacity/self-driving-car. Accessed: Jun. 2018.*