# CarND-Path-Planning-Project

In this project, your goal is to design a path planner that is able to create smooth, safe paths for the car to follow along a 3-lane highway with traffic. A successful path planner will be able to keep inside its lane, avoid hitting other cars, and pass slower moving traffic all by using localization, sensor fusion, and map data.

The project submission contains the following files:

- main.cpp: **all the code changes have been made to this file**
- helpers.h: not modified
- spline.h: **added**
- json.hpp: not modified
- CarND-Path-Planning-Project.pdf: writeup
- New video_Udacity.mp4

The following criteria were taken from the project rubric, for implementation:

- The car should try to go as close as possible to the 50 MPH speed limit without exceeding it.
- The car should Pass slower traffic, as required and when it is safe to change lanes
- The car should avoid hitting other cars
- The car should always drive within the marked road lanes, unless going from one lane to another.
- The car should be able to make one complete loop around the 6946m highway.
- Also the car should not experience total acceleration over 10 m/s^2 and jerk that is greater than 10 m/s^3.

The map of the highway is in data/highway_map.csv

Each waypoint in the list contains [x,y,s,dx,dy] values.

- x and y are the waypoint's map coordinate position
- the s value is the distance along the road to get to that waypoint in meters
- the dx and dy values define the unit normal vector pointing outward of the highway loop. The highway's waypoints loop around so the frenet s value, distance, goes from 0 to 6945.554.

Reflection:

The project code in main.cpp is broken into smaller segments:

1. Prediction: I tried a couple different approaches.
    - Use the sensor fusion data to determine if there are:
        i. Car in front (tooClose)
        ii. Car in the left lane (leftLaneOccupied)
        iii. Car in the right lane (rightLaneOccupied)

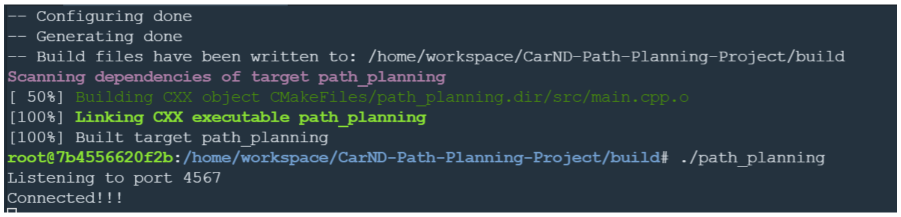
- We consider a safety window of 30 m in front of or behind us while making lane change determinations

2. Behavior:
    - To determine slow down or speed up or do a lane change
        i. lane -= 1 ➔ left side lane change
        ii. lane =+ 1 ➔ right side lane change
        iii. ref_vel += 0.224➔ accelerate in the same lane till we reach 49.5 MPH
        iv. ref_vel -= 0.224➔ decelerate in the same lane
3. Trajectory Planning:
    - This code calculates the trajectory based on the speed and lane output from the behavior, car coordinates and past path points.
    - Spline initialization: use the last 2 points of the previous trajectory or the car position

I used the following method to build and compile the code, from the Udacity workspace:

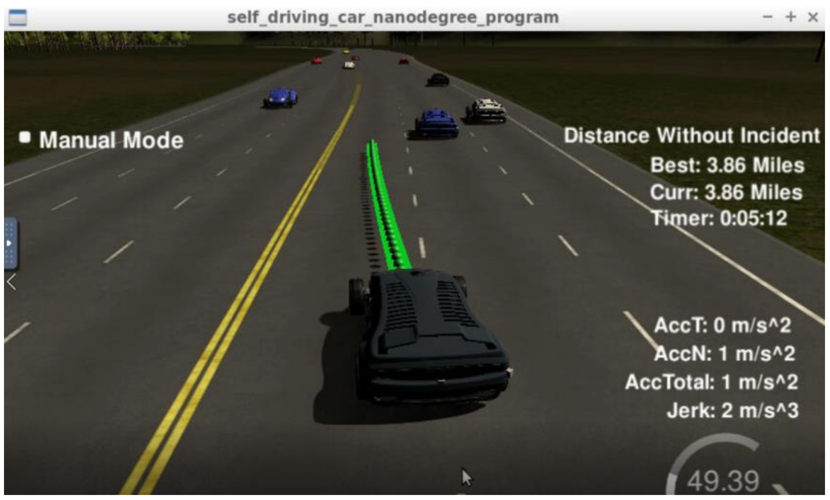
- mkdir build && cd build **to create and enter the build directory**
- cmake .. && make **to compile the project**
- ./path_planning **to run the code**

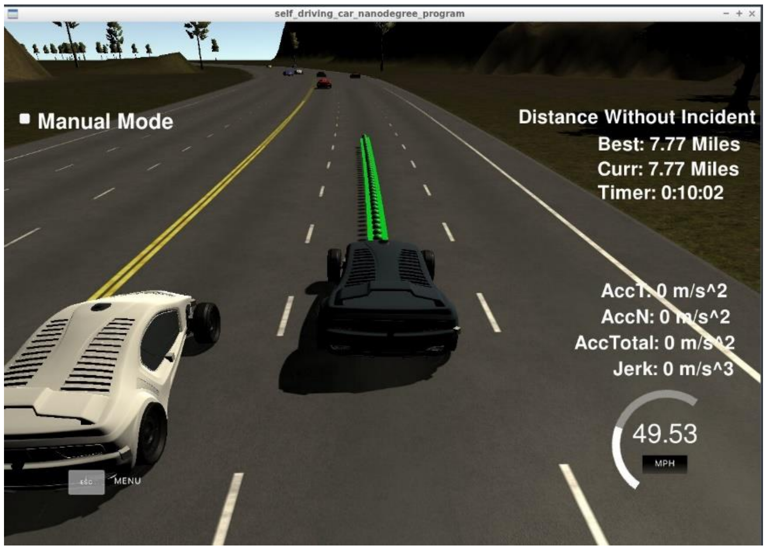
To see the results on the simulator, enable GPU mode and Click on the "Simulator" button in the bottom of the Udacity workspace, which will open a new virtual desktop. You should see a "Simulator" icon on the virtual desktop. Double-click the "Simulator" icon in that desktop to start the simulator with the code running on the terminal.



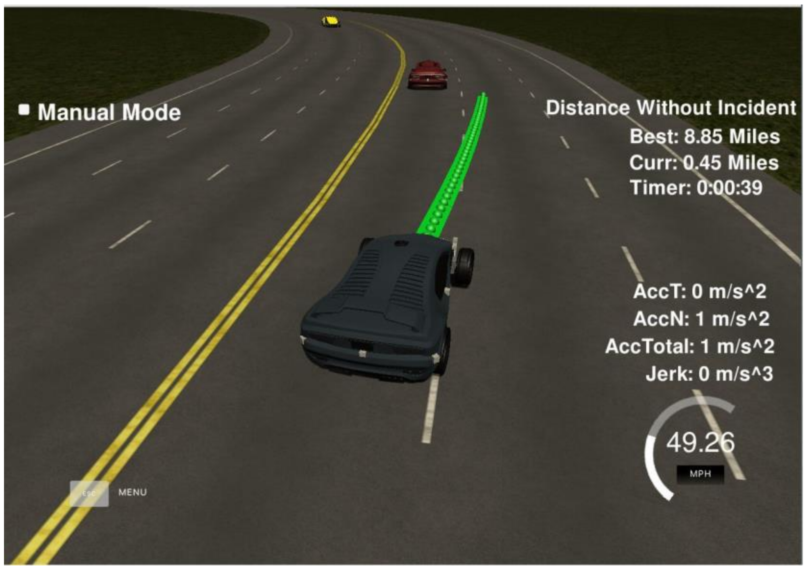
```
-- Configuring done
-- Generating done
-- Build files have been written to: /home/workspace/CarND-Path-Planning-Project/build
Scanning dependencies of target path_planning
[ 50%] Building CXX object CMakeFiles/path_planning.dir/src/main.cpp.o
[100%] Linking CXX executable path_planning
[100%] Built target path_planning
root@7b4556620f2b:/home/workspace/CarND-Path-Planning-Project/build# ./path_planning
Listening to port 4567
Connected!!!
```

Below screenshots show the car driving on the simulator





Possible events: Yes. I did see collisions/ events eventually. I guess there needs to be additional cost factors or improvements to the state machine or increased safety window limits, but this was after the requirements for the project were met.


Manual Mode
Distance Without Incident
Best: 8.85 Miles
Curr: 0.45 Miles
Timer: 0:00:39
AccT: 0 m/s^2
AccN: 1 m/s^2
AccTotal: 1 m/s^2
Jerk: 0 m/s^3
49.26
MPH
MENU