Udacity SDCND

Path Planning Project Reflection *By Daliso Zuze*

The code compiles correctly.

- I was able to successfully compile and execute the code. I generated and Xcode profile and used Xcode as my IDE for development of this project.

The car is able to drive at least 4.32 miles without incident.

- The car was able to drive the full lap without incident, please see video recording here: https://youtu.be/TWKqvPf25KY

The car drives according to the speed limit.

- The car did not violate the speed limit during my trial runs. Please refer to the video as an example: https://youtu.be/TWKqvPf25KY

Max Acceleration and Jerk are not Exceeded.

- The car did not violate max acceleration or jerk during my trial runs. Please refer to the video as an example: https://youtu.be/TWKqvPf25KY

Car does not have collisions.

- The car did not have a collision during my trial run. Please refer to the video as an example: https://youtu.be/TWKqvPf25KY

The car stays in its lane, except for the time between changing lanes.

- The car remained in its lane except for during lane changes. Please refer to the video as an example: https://youtu.be/TWKqvPf25KY

The car is able to change lanes

- The car was able to change lanes when it was safe to do so. Please refer to the video (time: 55s) as an example: https://youtu.be/TWKqvPf25KY

There is a reflection on how to generate paths.

My path planning code communicates with the term 3 simulator to continually feed it with upcoming xy path points for it to follow. I decided to follow the approach demonstrated in the class walk-through by supplying the simulator with 50 points during each loop through the path planning code. The simulator is designed to move the car to the next waypoint in the path every .02 seconds, therefore 50 path points represents 1 second of travel. The spacing of these points represents how fast the car is moving.

The 50 points are generated as follows:

- The first step is to determine the velocity we are targeting and the lane that the vehicle should be in. We initialize lane to 1 (line 237) and velocity to 0 (line 240)

- At the start of each path generating loop the planner receives telemetry from the vehicle. It uses some of this telemetry to analyse the state of the road around it. (Line 292)
- If the road ahead is clear, and the car's previous target velocity was below the speed limit then we keep the lane unchanged and increase the target velocity by a small amount, 0.224 mph , this amount is small enough to prevent the vehicle from exceeding the maximum acceleration and jerk constraints.
- If the road ahead is not clear then the car either changes lanes if possible or slows down to prevent a collision with a vehicle in front of it. (Lines 315-335)
- Once we have the target velocity and the lane determined, we go ahead to generate the points.
- An important point in the approach I used is that the car's previous path is allowed to complete, I only add on additional points. In order to do this I generate a smooth path forward proceeding from where the previous 50 points ended.
- This is achieved by generating 5 anchor points for a spline . The first 2 anchor points are the last 2 points of the previous path. The remaining three anchor points are 30,60, 90 meters ahead in the lane we are planning the car to be in. (Line 346-404)
- Once we have the spline we then just need to pick points on the spline evenly spaced such that the spacing corresponds to the target velocity we set earlier.(lines 437-466)
- During the normal running of the code with the simulator, we will only be adding approximately 2 points to the overall path of the vehicle, the first 48 points would be carried over from the last iteration through the planner. If we increase the target velocity each iteration this is what gives us our smooth gradual increase in velocity.