Model Documentation:

The aim of the project is to generate a smooth path/trajectory for the car running on a highway.

There are three main steps involved. They are 1) Obstacle Detection with Sensor Fusion, 2) Calculate safest lane and velocity that the car must maintain and 3) Generate a smooth trajectory from current lane to target lane by following the safe velocity calculated above.

First, as a note, the highway scenario has three lanes for the onward traffic. In Frenet Co-ordinates the distance of the car from the start of the left most lane is represented by **"d". "s"** in Frenet co-ordinates represent the distance that car has travelled on the highway.

Also, a trajectory that the car can take for next one second is calculated every 0.02 seconds. So, the trajectory will have 50 points.

1) Obstacle Detection with Sensor Fusion:

Onboard sensors on car track all the obstacle vehicles around the car. And they provide the **"x, y"** co-ordinates of the car along with the corresponding **"s, d"** values in Frenet. Also, the obstacle's velocity along x and y components are also available, using which we can calculate the magnitude velocity of the obstacle vehicle. Using the obstacle velocity, we can calculate the obstacle's **"s"** with respect to the distance our car could also travel in next few seconds, so we can avoid collision. Thus, if a vehicle is 30 meters ahead or 20 meters far to our vehicle's left or right side it is considered an obstacle and is recorded.

Thus, our car is aware of the obstacles around it.

The method **"check_if_vehicles_around()"** on **"helpers.h"** does this calculation.

2) Calculate safest lane and velocity for the car:

From the records we took in step one, our car can now decide the safest lane while trying to maintain a 50mph speed. If an obstacle is found ahead, the car prefers to change to left lane, which if not possible tries to shift to right lane and as a last resort it tries to slow down. The vehicle maintains its lane and tries to maintain 50mph if it has no obstacle in front.

These calculations are performed by the method **"get_resultant_speed_and_lane()"** on **"helpers.h".**

3) Generate a smooth trajectory:

As mentioned previously, the car tries to calculate a safe trajectory that it can take for next one second. But, in next 0.02 seconds it again calculates the safest trajectory, by which time it would not have passed all the 50 points previously generated. So, using these left out previous points and the newly calculated target lane and speed a smooth trajectory is generated using **"spline"** library.

These calculations are performed by the method **"generate_smooth_trajectory()"** on **"helpers.h".**