Project: Path-Planning

Foreword:

I first took a look at the structure of the code, the signatures of the object, Sensor-fusion etc. I followed ever step from the project Q&A. I will explain the code in steps:

Step1: make sure the car moves 8 (are commented out)

I used the code part from the getting started (lines *119- 123*) to make sure the car moves at all

Step2: make the car move in a defined lane (are commented out)

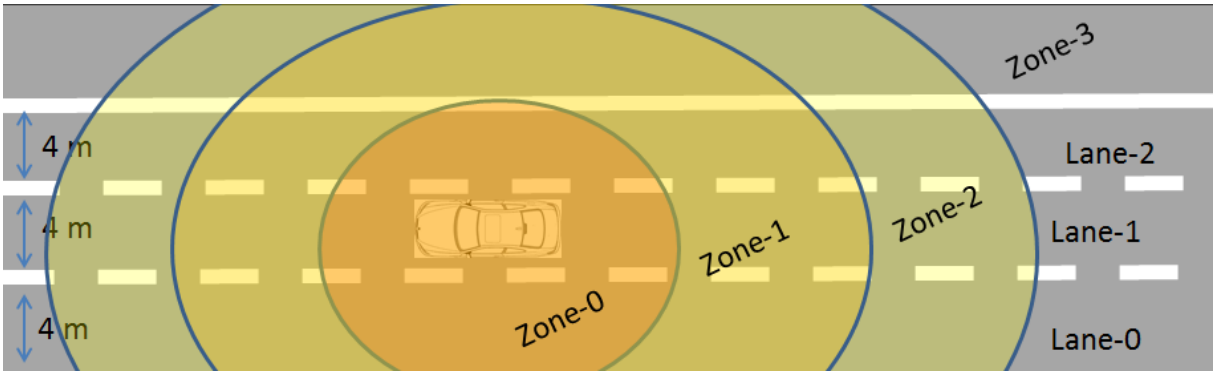
Also used the code from the instruction to make the car move in a defined lane in the freenet coordinate (lines *131-138*)

Step3: use the logic explained in the project Q&A to create an array of x and y coordinates for the car to follow (lines *413-525*)

- Start from the current position
- Create an array to hold the previous and the current x,y values
- Create a tangent using the previous two x-, y- coordinates (lines *421-452*)
- Using this create 3- points spaced at 30m, 60m and 90m from the reference (lines *455-465*)
- Transform the x-, y- coordinates with reference to the car (lines *467-475*)
- Set x-, y- points to spline to smoothen the path (line *481*)
- Break up spline points in segments of 30m to match the desired velocity (lines *495-531*)
- Transform the x-, Y- coordinates back to global coordinates (lines *516-517*)
- Add them to the x-, y- coordinates to define the next x- and y-
- Create next_x_vals and next y_valy vectors (lines *522-523*)
- This will be sent to the simulator through Json

Step4: Path planning strategy (lines *164-402*)

- Get the sensorfusion values in a for loop (lines 181)
- Get a sense for the relative speed of the ith car in the loop with respect to our car, to assess danger of collision (lines *195-220*)
    - If the ith car is coming from behind and has a positive relative speed
    - If the ith car is in front and has a negative relative speed
    - In the opposite cases there is no immediate danger
- Get a sense of the distance of the ith car wrt our car (lines *226- 249*)



    - Zone-3: vehicles here are at a safe distance away
    - Zone-2: vehicles are at a safe distance but have to be observed
    - Zone-1: vehicles are dangerously close, lane change is possible but not preferred
    - Zone-0: vehicle is dangerously close, braking is a priority, lane change is not a priority
- Assign cost factors for turning left or right based on (line *270- 349*)
    - The zone in which a vehicle has been detected
    - The assessment of relative speed
- Intervention detected (line 379)
- Course of action (lines *380-402*)
    - Assign swerving left if it can be safely done
    - Else: assign swerving right if it can be safely done
    - If lane change is not possible decelerate at a rate based on the zone
    - If no course of action is desired, accelerate back to the desired speed

Conclusion:

The vehicle drives at the desired speed on the desired lane, if an intervention is required it changes lane when possible, brakes if a lane change intervention is not possible. The car drives aggressively and does crash into other cars sometimes (Tries to find a path between cars and so on). After a lot of tweaking with the weights it was possible to meet all criteria. With some more tuning I'm sure this code can make the car drive perfectly

If during your trial it fails and it can't be accepted, I will upload another rudimentary code I had developed earlier. With this code the car does not drive aggressively and change lane when it is very safe, else it brakes. No state machine or weights are incorporated there and it definitely satisfies all criteria all the time, but was not satisfactory for me.