**Project Specification – Path Planning**

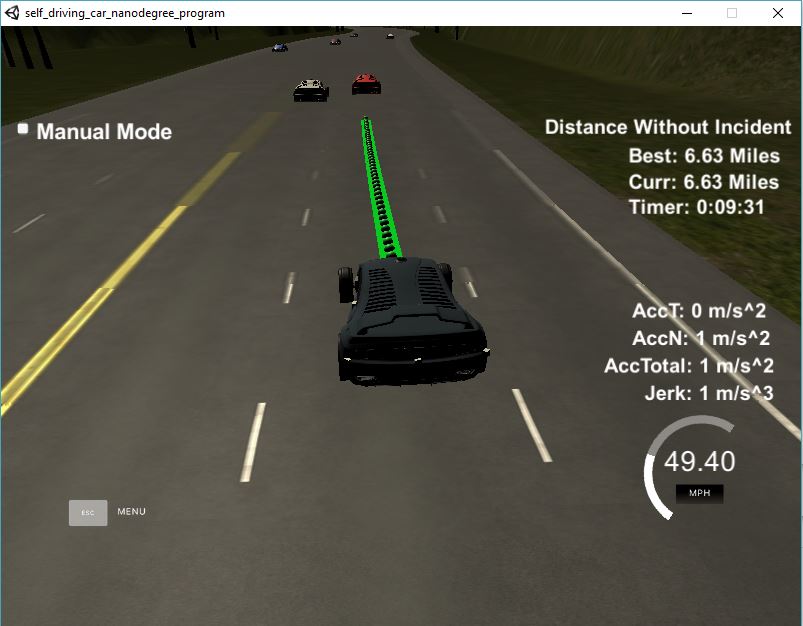
**Compilation**

**The code compiles correctly**

The code compiles correctly with cmake. You might have to delete CMakeCache.txt under /build for the cmake to refresh according to your directory.

**Valid Trajectories**

**The car is able to drive atleast 4.32 miles without incident**



As seen in image the car was able to drive 6.63 miles without any incident. The car can drive over 15 miles without an incident as per several tests conducted.

**The car drives according to the speed limit.**

The car drives with a maximum speed set to 49.5 mph which is less than the speed limit of 50 mph and never exceeds that speed

**Max Acceleration and Jerk are not Exceeded.**

The car never exceeds the maximum acceleration and jerk even during the start.

**Car does not have collisions**

The car does not have collisions and will remain maintain a safe distance from all vehicles except if a car changes lane in front of it at the last moment as noticed during one of the test runs.

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

The car will stay its lane and maintain that unless required to change lanes

**The car is able to change lanes**

The car can change lanes smoothly whenever it is safe to do so.

**Reflection**

The code can be divided into the following parts:

1. Prediction
2. Behavior Planning
3. Trajectory Generation

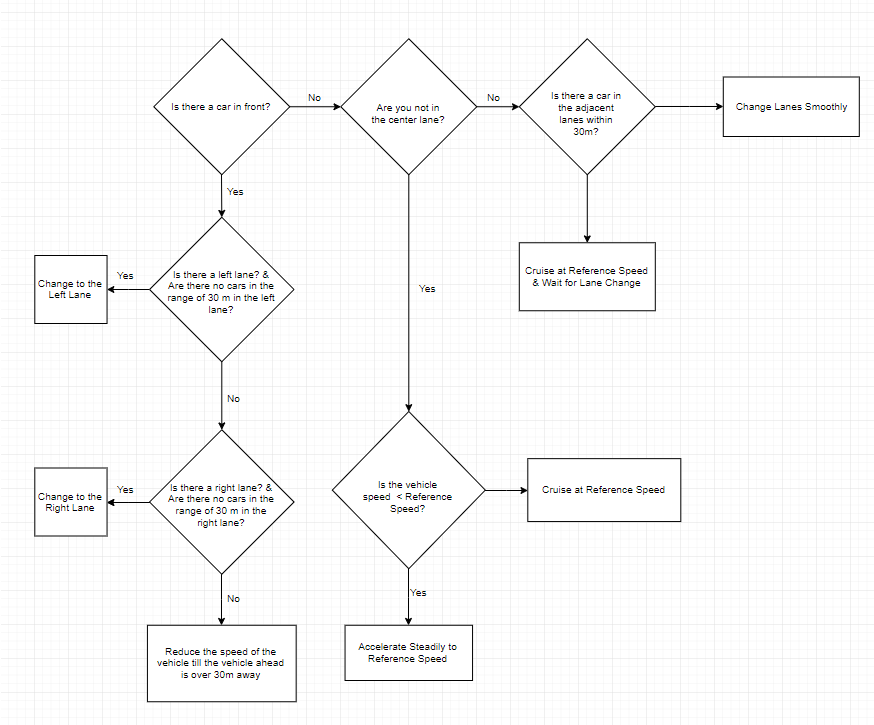
**Prediction (Lines 248 to 306)**

This part calculates the position of other cars with respect to the host car. The code uses the sensor fusion data to locate the vehicles. The first part (lines 248 to 288) defines the lane of the car using the lateral position of the vehicle. The lateral position (d), obtained from the sensor fusion data, is considered positive for vehicles travelling in the same direction and negative for vehicles in the opposite direction. Considering a lane is 4 m wide, the lane indexes of the vehicles are calculated for the vehicle with positive indexes for same direction lanes and negative for opposite lanes. The left most lane in the direction of travel is considered as the 0th lane.

The second part (lines 289 to 306) defines the longitudinal distance of the vehicles from the host vehicle. This defines the margin of safety the vehicle must maintain always. This is defined to be 30 m in the code. The margin of safety is defined by the answer to the question: Is the car within a 30 m range of the host vehicle. Vehicles behind the host in the same lane are not considered in this condition. So, the vehicle defines an area of safety 30 m in the same lane ahead of it. This area of safety is used in the planning the behavior of the vehicle.

**Behavior Planning (Lines 308 to 339)**

The following flowchart will explain the behavior planning used:



**Trajectory Generation (Lines 341 to 447)**

The first part of the code uses the last two points of the previous trajectory (if it exists) and 3 points at 30, 60 and 90 meters ahead of the vehicle that depend on the lane index of the vehicle and the map waypoints data to generate a spline. The spline is generated using the ‘spline.h’ library. Using the last two points of the previous trajectory ensures continuity of the spline function. Next the points and convert to the vehicles local co-ordinates to maintain the origin of the spline at (0,0). Based on the speed of the vehicle a total of 50 points are generated along the spline using a technique discussed in the walkthrough that incorporates the effect of speed on the trajectory. Finally the points are converted back to global co-ordinates and pushed for the vehicle to follow.