Path Planning Project

Compilation

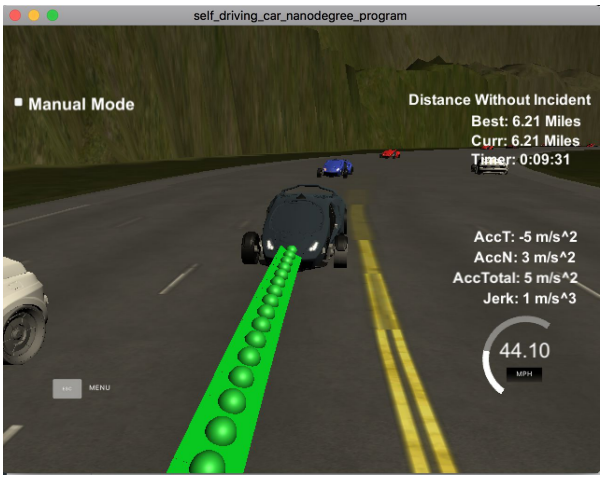
Once the install for uWebSocketIO is complete, the main program can be built and run by doing the following from the project top directory.

1. ./build.sh - build project
2. ./clean.sh - clean build folder
3. ./run.sh - connect pid to Term3 simulator

Valid Trajectories

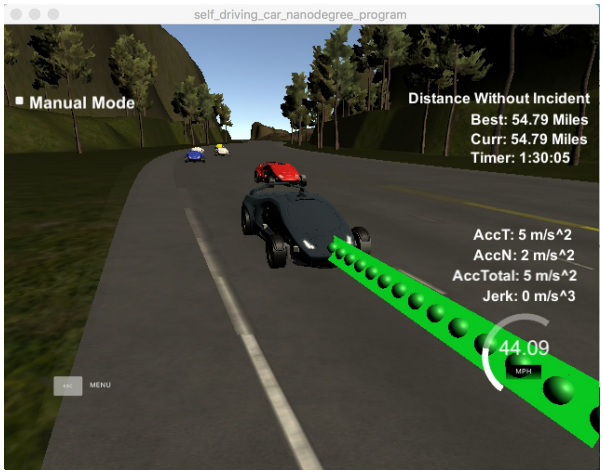
The car is able to drive at least 4.32 miles without incident..	The car drives 50 miles without collisions.
The car drives according to the speed limit.	The car doesn't drive faster than the speed limit.
Max Acceleration and Jerk are not Exceeded.	The car does not exceed a total acceleration of 10 m/s^2 and a jerk of 10 m/s^3.
Car does not have collisions.	The card does not come into contact with any of the other cars on the road.
The car stays in its lane, except for the time between changing lanes.	The car doesn't spend more than a 3 second length outside the lane lanes during changing lanes, and every other time the car stays inside one of the 3 lanes on the right hand side of the road.
The car is able to change lanes	The car is able to smoothly change lanes when it makes sense to do so, such as when behind a slower moving car and an adjacent lane is clear of other traffic.

Results


self_driving_car_nanodegree_program
Manual Mode
Distance Without Incident
Best: 6.21 Miles
Curr: 6.21 Miles
Timer: 0:09:31
AccT: -5 m/s^2
AccN: 3 m/s^2
AccTotal: 5 m/s^2
Jerk: 1 m/s^3
44.10
MPH
MENU


self_driving_car_nanodegree_program
Manual Mode
Distance Without Incident
Best: 8.62 Miles
Curr: 8.62 Miles
Timer: 0:13:00
AccT: 0 m/s^2
AccN: 0 m/s^2
AccTotal: 0 m/s^2
Jerk: 0 m/s^3
49.42
MPH
MENU


self_driving_car_nanodegree_program
Manual Mode
Distance Without Incident
Best: 12.94 Miles
Curr: 3.21 Miles
Timer: 0:05:00
AccT: 5 m/s^2
AccN: 1 m/s^2
AccTotal: 5 m/s^2
Jerk: 0 m/s^3
37.54
MPH
MENU


self_driving_car_nanodegree_program
Manual Mode
Distance Without Incident
Best: 54.79 Miles
Curr: 54.79 Miles
Timer: 1:30:05
AccT: 5 m/s^2
AccN: 2 m/s^2
AccTotal: 5 m/s^2
Jerk: 0 m/s^3
44.09
MPH
MENU

Reflection

Prediction (252 - 289 lines)

We get information from sensor fusion data and find a free and safe lane. The lane will be marked as free if there is no car within 30 meters in front of or behind the vehicle.

Behavior planning

This part decides to change speed or lane based on predicted data. If there is a car ahead of the vehicle and it is not blocked, it will change lane to left or right one. This is a very simple planner, it can allow only single lane switch without double-crossing, so the vehicle could be potentially be blocked by two cars in front, there is a place for improvement.

Trajectory generation (320 - 397 lines)

Based on the speed and the lane from the behavior part this code will generate trajectory.

The spline needs initial coordinates (327 - 346 lines) and final coordinates which were built with the help of planner lane output (348 - 354 lines).

We need coordinates transformation (362 - 368 lines) before evaluating coefficients of the trajectory polynomial.

After previous steps, we are ready to add new values to next_x_vals and next_y_vals.
At first, we will copy previous path coordinates (372 - 378 lines) and then we will generate next points (379 - 396) with the help of spline and non-local transformation.

The speed calculation part (384 - 389 lines) changes the speed on every trajectory points instead of doing it for the complete trajectory.