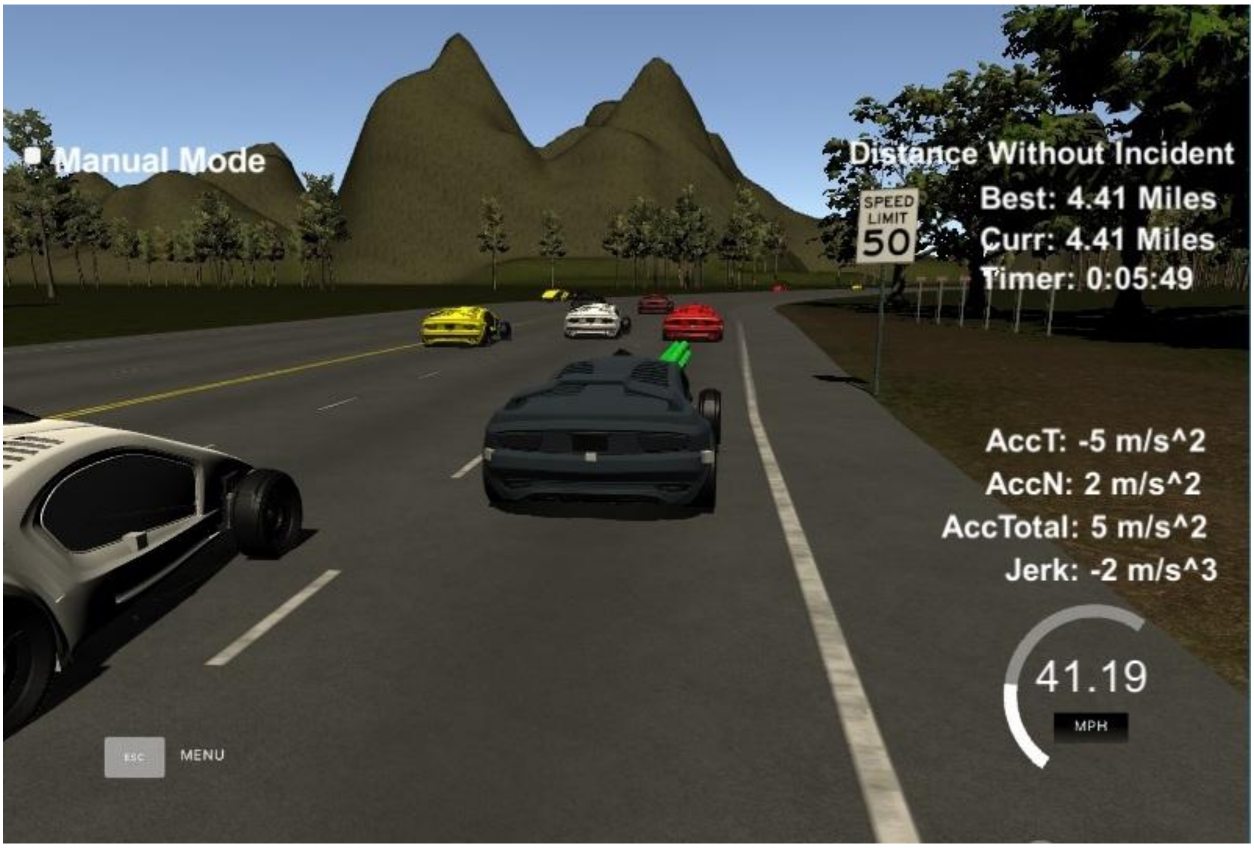
Path Planning Project



In this project my goal was to write a program that can find the best trajectory for the car on the road that also contains other cars.

Processing data from the car's sensors

In lines 267-321 there is a loop that processes the data from car's sensors. From that data I can detect cars on the road and learn their parameters: lane number, speed, future s coordinate and their position regarding the ego car.

Path planning

In latter lines (323-443) the car can change the lane if it has a car ahead on the same line and there is no cars that are too close to the ego car on the nearest lane. The ego car also prefer to choose the central lane if it is empty.

From sensor data I also gathered the future s coordinates of the cars that are not too close but also going ahead of the ego car on each lanes (their s coordinates are getting added to the vectors *cars_center_forward_v, cars_right_forward_v,* and *cars_left_forward_v*). I use these s coordinates to decide what to do in these quite rare situations:

1. which lane should the ego car take if it is on the center lane and has a car ahead and both side lanes do not have cars that are close to the ego car. The car should observe the car far behind and choose the lane that has farthest car (lines 334-357, 360-395).
2. should the ego car go to the center lane if it has a car on it that is not too close but in the future became an obstacle (lines 397-436).

Speed and acceleration control

In lines 439-442 I correct the ego car's speed to maintain it near 49.5 miles per hour but not more than that. I also prevent car's acceleration from increasing over 0.224 m/s^2.

Getting smooth trajectory

For smoothing a trajectory I used the spline library that used path points converted from map to local coordinates (lines 491-503).