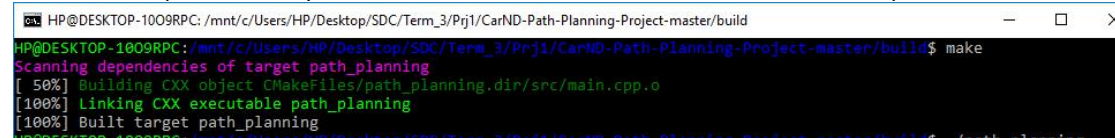


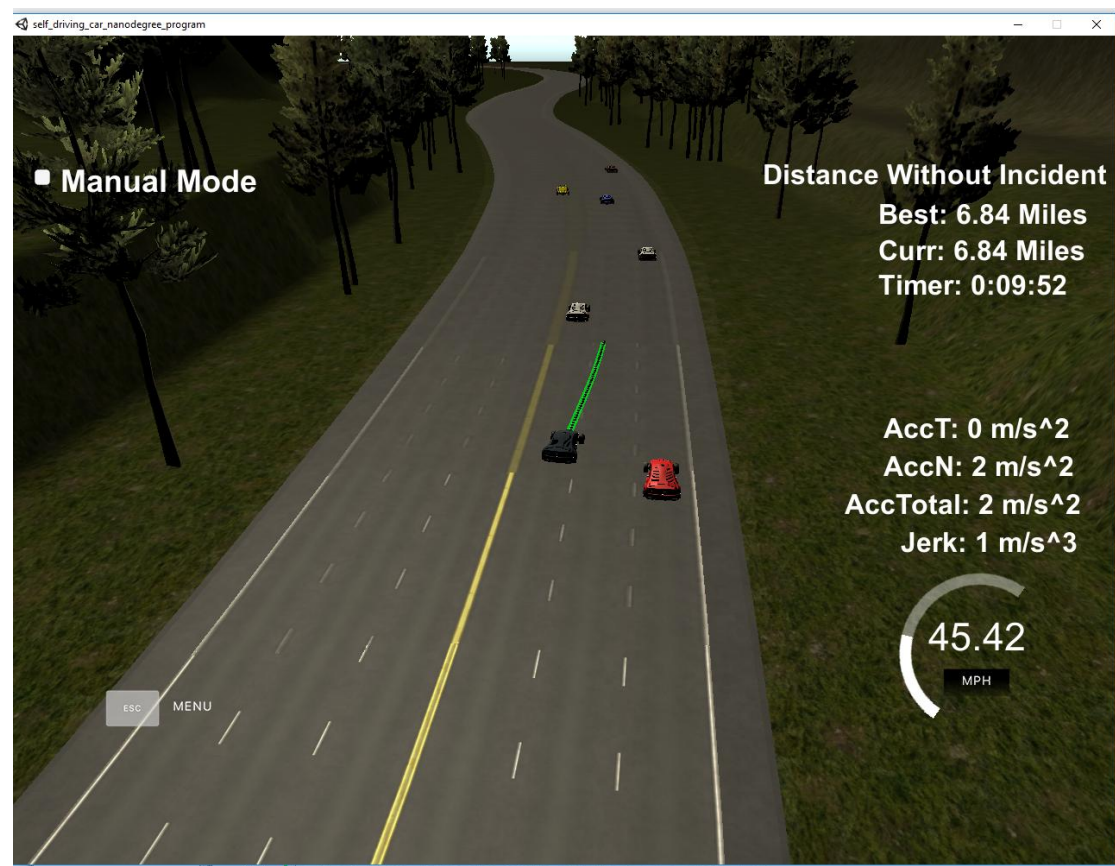
Compilation and self-driving

The code compiles correctly. There is no error when use cmake and make to compile the files in src.



```
HP@DESKTOP-1009RPC: /mnt/c/Users/HP/Desktop/SDC/Term_3/Prj1/CarND-Path-Planning-Project-master/build$ make
Scanning dependencies of target path_planning
[ 50%] Building CXX object CMakeFiles/path_planning.dir/src/main.cpp.o
[100%] Linking CXX executable path_planning
[100%] Built target path_planning
```

The car is able to drive at least 6.84 miles without incident. See the following screen shoot. And the car drives according to the speed limit. I set the limit to 45 mph.



During the run, max acceleration and jerk are not exceeded. And car does not have collisions. Also the car stays in its lane, except for the time between changing lanes. As shown above the car changes lane when there it is proper.

Reflection of path generation

To generate the path, please look at line 254-289. First , we need to find the cars near our car. With that information, we can get to know whether there is a car ahead, on the near left behind and right behind.

From line 292-309, we defined the rule of action if there is a car ahead of our car. First we will try to see whether the nearby left lane is occupied, if not we will merge to the left. If the left lane is

not available, we then go to see whether the right lane is available. If both left and right lane are not available, we will just slow down our car by putting the desired decrease of speed into Speed_diff. At last, if we find that no car is ahead of us, we may want to accelerate our car. So an additional speed increment is put into Speed_diff. Later we will use this parameter Speed_diff to decide the reference speed of the car.

From line 401-407, as mentioned in the last paragraph, we will change the reference speed of our car by adding the Speed_diff into the ref_vel. If the resulting ref_vel is exceeding the speed limit, will just set it equal to speed limit, if the resulting ref_vel is too small, then make it into smallest speed increment.