UDACITY

< Return to Classroom

DISCUSS ON STUDENT HUB

# Path Planning

REVIEW

CODE REVIEW 1

HISTORY

## Meets Specifications

Future SDC engineer,

You have done a great job on this project! You understand all parts of this project well. I enjoyed reviewing your work.

**Here are some links which I liked going through after finishing the project:**

- Introduction to Robotics #4: Path-Planning
- The path planning problem in depth
- A discussion on What is the difference between path planning and motion planning?
- Introduction to robot motion: Robot Motion Planning
- Introduction to robot motion: Path Planning and Collision Avoidance

## Compilation

**Code must compile without errors with `cmake` and make.**

**Given that we've made `CMakeLists.txt` as general as possible, it's recommend that you do not change it unless you can guarantee that your changes will still compile on any platform.**

Awesome Job in this section!! The code compiled without errors with cmake and make when running from

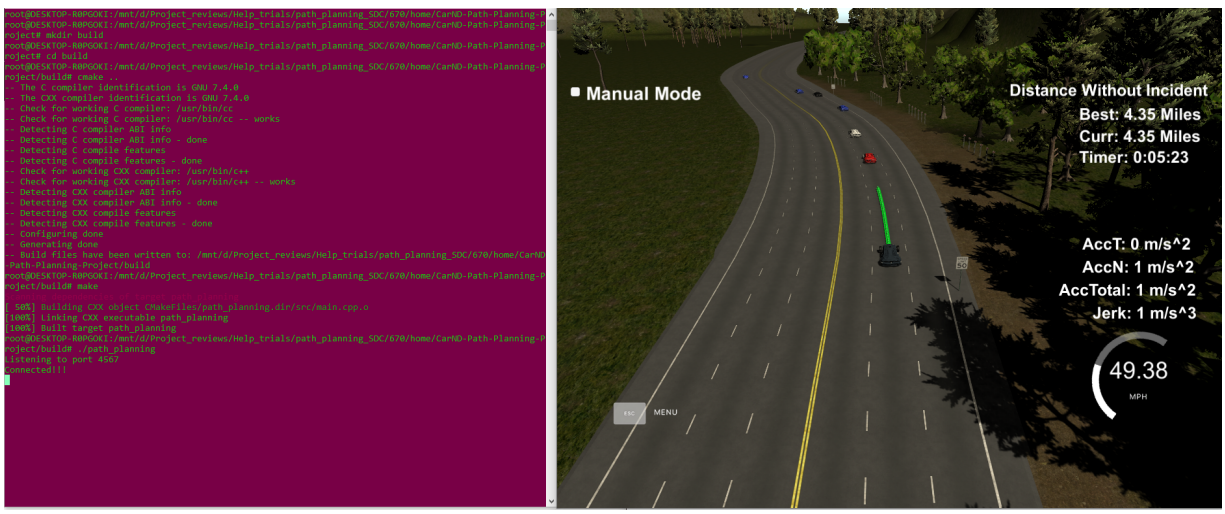
the build directory. Great keep it up!

```
project/build# cmake ..
-- The C compiler identification is GNU 7.4.0
-- The CXX compiler identification is GNU 7.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/Project_reviews/Help_trials/path_planning_SDC/670/home/CarND
-Path-Planning-Project/build
root@DESKTOP-R0PGOKI:/mnt/d/Project_reviews/Help_trials/path_planning_SDC/670/home/CarND-Path-Planning-P
roject/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
```

## Valid Trajectories

**The top right screen of the simulator shows the current/best miles driven without incident. Incidents include exceeding acceleration/jerk/speed, collision, and driving outside of the lanes. Each incident case is also listed below in more detail.**

Well done in this section! The car passes the requirement of `4.3` miles without incidents.




**The car doesn't drive faster than the speed limit. Also the car isn't driving much slower than speed limit unless obstructed by traffic.**

Well done in this part. The car adjusts the speed automatically. It adjusts the speed based on the nature of

traffic.

Just that you can improve the logic of the car even further. When there is no chance of changing the lanes your car stays in the lane but accelerates and decelerates. Instead, your car can adapt the speed of the car ahead of you in that lane. That way the ride will be smoother and you will have more chances of changing the lanes and overtaking the other cars.

**The car does not exceed a total acceleration of 10 m/s^2 and a jerk of 10 m/s^3.**

Yes, the car does not exceed the max acceleration and max jerk. It drove safely throughout the track.

## Some links to get more insights

Path Planning for Collision Avoidance Maneuver
Optimal Trajectory Planning for Glass-Handing Robot Based on Execution Time Acceleration and Jerk
This discussion on StackExchange can be of interest Which trajectory planning algorithm for minimizing jerk

**The car must not come into contact with any of the other cars on the road.**

**The car doesn't spend more than a 3 second length out side 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 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.**

# Reflection

**The code model for generating paths is described in detail. This can be part of the README or a separate doc labeled "Model Documentation".**

A beautiful write-up. You explained everything in detail. I enjoyed going through it.

DOWNLOAD PROJECT

1 CODE REVIEW COMMENTS

RETURN TO PATH