

## PROJECT

## Path Planning

A part of the Self-Driving Car Program

PROJECT REVIEW

CODE REVIEW

NOTES

## Meets Specifications

SHARE YOUR ACCOMPLISHMENT



Enthusiastic Learner,

This work is one good piece of a well-engineered project. Continue in this spirit and sure will skyrocket in this area. 😊

Continue reading smart student .....

## Further Improvement Suggestions

- To expand ones knowledge in this area, it might be good in ones extra time to look at the following materials:
  - [Introduction to Robotics #4: Path-Planning](#)
  - <http://www.coppeliarobotics.com/helpFiles/en/pathPlanningModule.htm>
  - The path planning problem in depth <https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume9/mazer98a-html/node2.html>
  - [http://www.roborealm.com/help/Path\\_Planning.php](http://www.roborealm.com/help/Path_Planning.php)

Rate this review

- A discussion on [What is the difference between path planning and motion planning?](#)
- [Excellent Tutorial on A\\* Robot Path Planning](#)
- [Path Planning in Environments of Different Complexity](#)
- [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`.

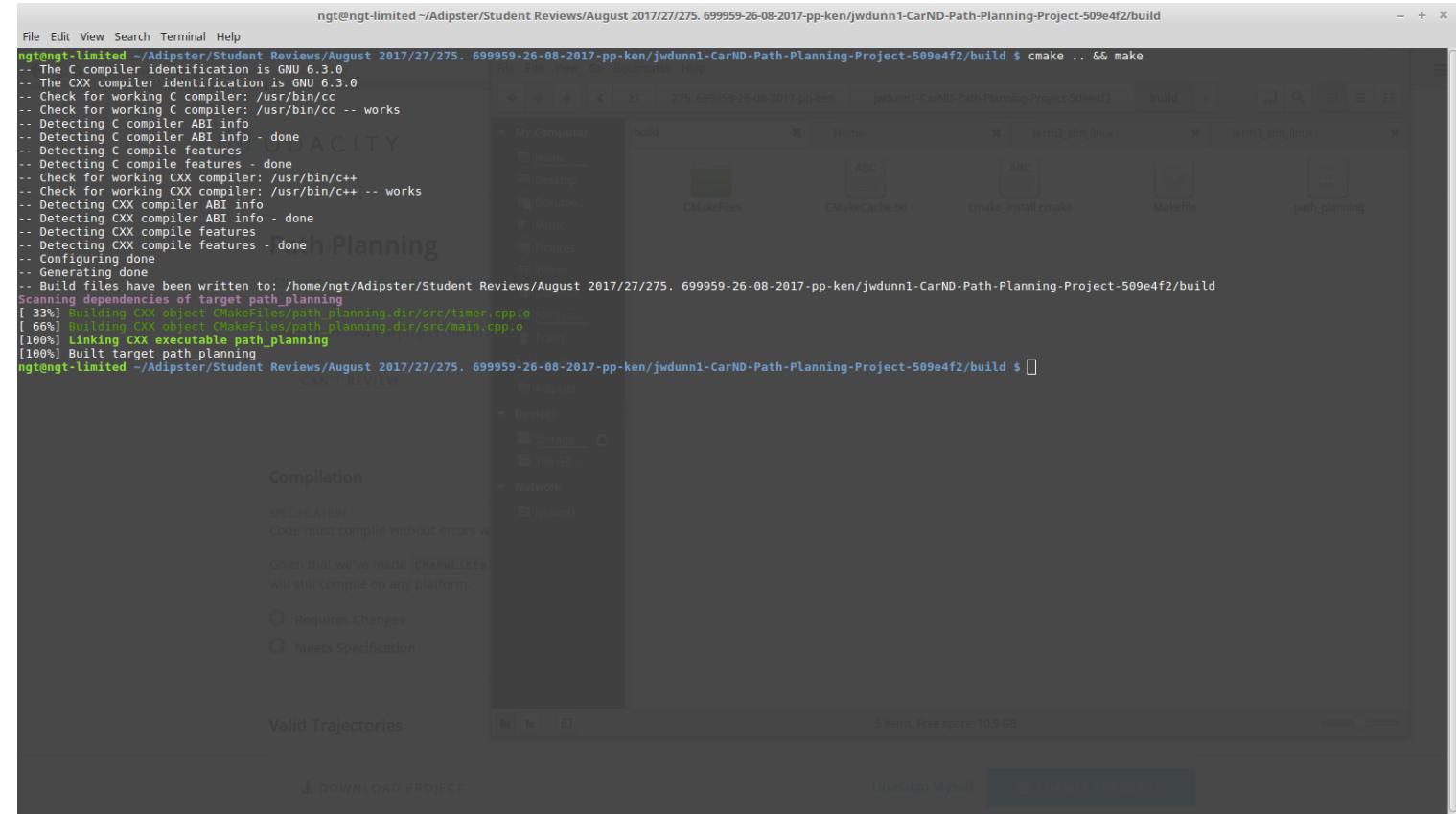
Given that we've made `CMakeLists.txt` as general as possible, it's recommend that you do not change it unless you can guarantee that your changes will still compile on any platform.

Impressive, keep it up!  The code compiled without errors with `cmake` and `make` when the command `cmake .. && make` is run from the build folder and with `CMakeLists.txt` submitted with the project.

## Further Improvement Suggestion

- Some C++ debugging tips to consider as a developer, see
  - [setting `CMAKE\_BUILD\_TYPE` to release](#)
  - [setting `CMAKE\_BUILD\_TYPE` to debug](#)
  - In case of any failure or difficulty, most commonly [Understanding why `CMAKE\_BUILD\_TYPE` cannot be set](#)
- When `cmake` includes symbol tables in the generated file, GDB now can come in to assist in debugging critical faults like the famous segmentation fault, why some variables are behaving weird, etc, see
  - [How to Debug Using GDB](#)
  - [GNU GDB Debugger Command Cheat Sheet](#)

- Debugging with GDB By Alexandra Hoffer



The screenshot shows a terminal window with the following text:

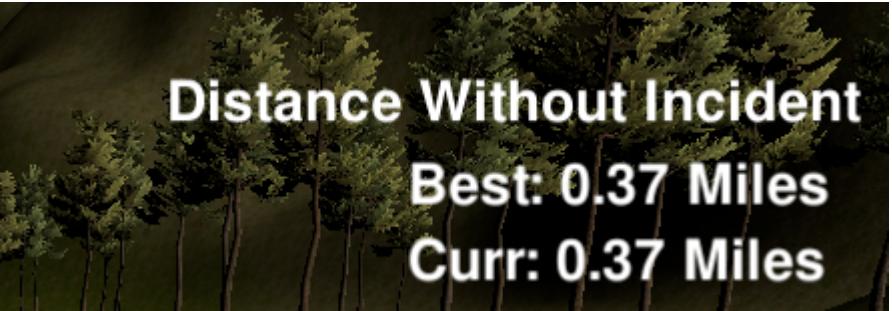
```
ngt@ngt-limited ~/Adipster/Student Reviews/August 2017/27/275. 699959-26-08-2017-pp-ken/jwdunn1-CarND-Path-Planning-Project-509e4f2/build
File Edit View Terminal Help
ngt@ngt-limited ~/Adipster/Student Reviews/August 2017/27/275. 699959-26-08-2017-pp-ken/jwdunn1-CarND-Path-Planning-Project-509e4f2/build $ cmake .. && make
-- The C compiler identification is GNU 6.3.0
-- The CXX compiler identification is GNU 6.3.0
-- 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 CXX 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: /home/ngt/Adipster/Student Reviews/August 2017/27/275. 699959-26-08-2017-pp-ken/jwdunn1-CarND-Path-Planning-Project-509e4f2/build
Scanning dependencies of target path_planning
[ 33%] Building CXX object CMakeFiles/path_planning.dir/src/timer.cpp.o
[ 66%] Building CXX object CMakeFiles/path_planning.dir/src/main.cpp.o
[100%] Linking CXX executable path_planning
[100%] Built target path_planning
ngt@ngt-limited ~/Adipster/Student Reviews/August 2017/27/275. 699959-26-08-2017-pp-ken/jwdunn1-CarND-Path-Planning-Project-509e4f2/build $
```

The terminal window is titled "jwdunn1-CarND-Path-Planning-Project-509e4f2". The background shows a file explorer window with a "build" folder containing files like CMakeLists.txt, CMakeCache.txt, cmake\_install.cmake, Makefile, and path\_planning.

## Valid Trajectories

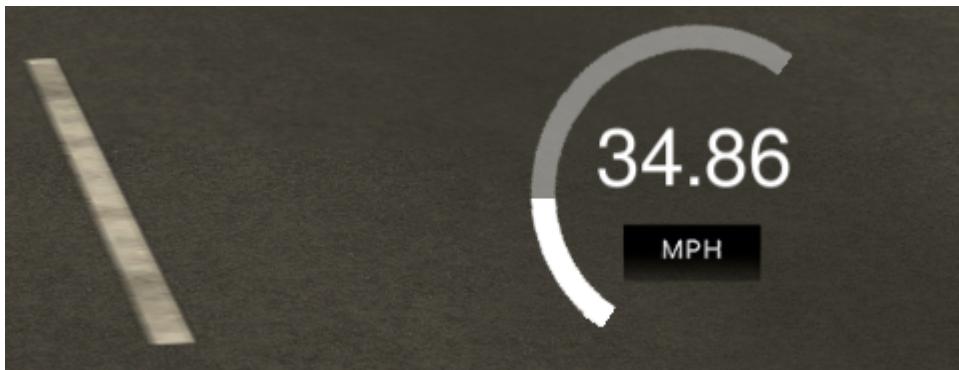
The top right screen of the simulator shows the current/best miles driven without incident. Incidents include exceeding acceleration/jerk/speed, collision, and driving outside of the lanes. Each incident case is also listed below in more detail.

Impressively done here. The top right screen of the simulator shows all the details as required. Outstanding performance with this visualization. 



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.

Awesome driving with auto speed adjustment. This work intelligently adjusts the speed based on the nature of traffic. This is outstanding work. The effort shows.



The car does not exceed a total acceleration of  $10 \text{ m/s}^2$  and a jerk of  $10 \text{ m/s}^3$ .

Interestingly, the car drives in a way if passengers were inside they will surely feel safe. It does not exceed total acceleration and jerk as required here. 😊

**AccT: 1 m/s<sup>2</sup>**

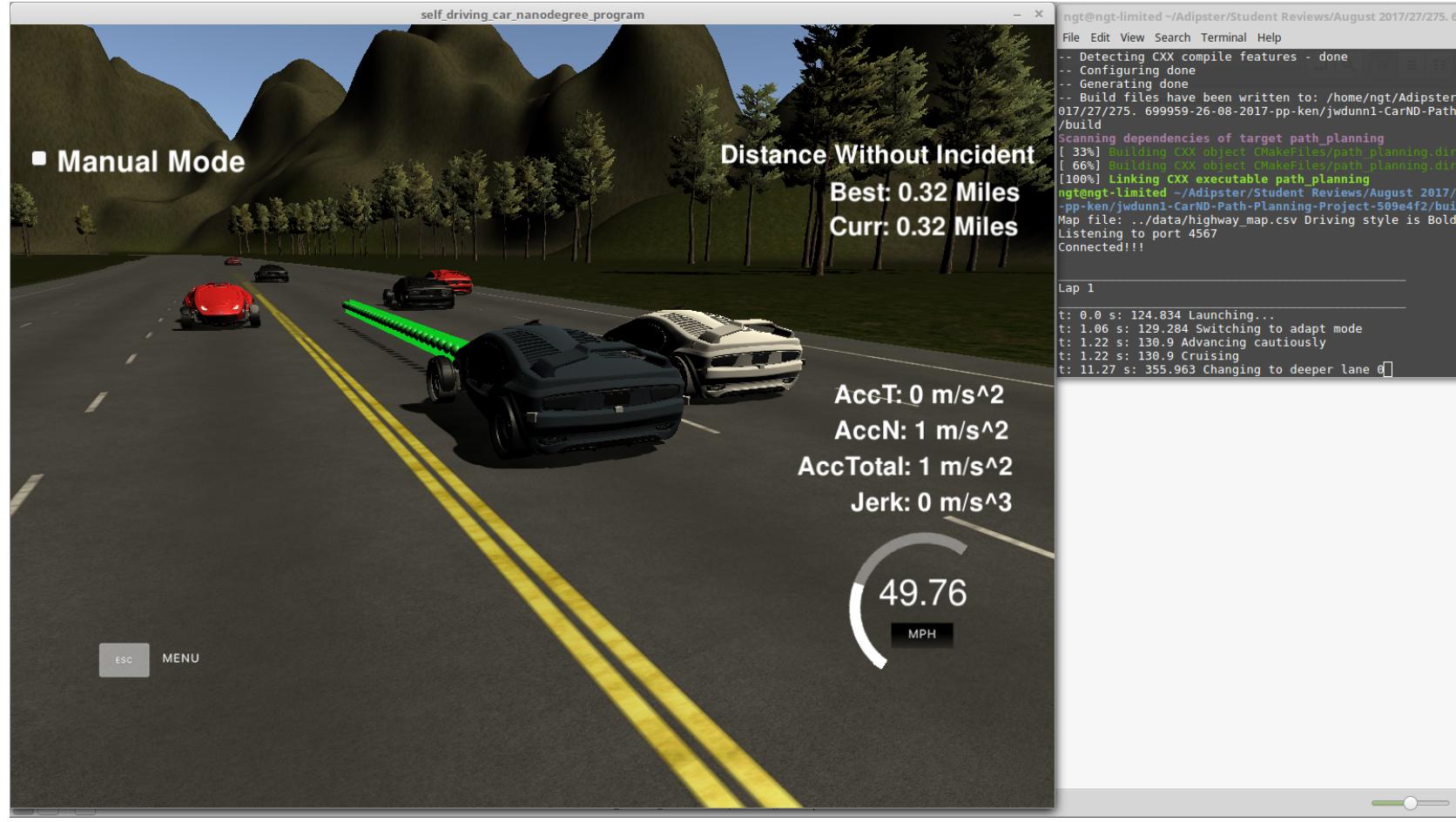
**AccN: 1 m/s<sup>2</sup>**

**AccTotal: 1 m/s<sup>2</sup>**

**Jerk: 1 m/s<sup>3</sup>**

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

The car drives through the whole miles without coming into contact with any other car. It smoothly switches lanes without bumping into other cars. I think this car is intelligent. One cannot tell if it is a self-driven car or it is a human inside. Keep it up, student! 🎉



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 easily and without spending much time in a traffic unnecessarily, change lane and does not delay outside lane lines during this change as expected. Furthermore, throughout the driving, the car stays inside one of the lanes on the right hand of the road. Impressive work. The effort shows.





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.

This car is the car needed out there. I am overwhelmed by this performance. Continue with such good implementations





```
self_driving_car_nanodegree_program
File Edit View Search Terminal Help
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ngt/Adipster/017/27/275. 699959-26-08-2017-pp-ken/jwdunn1-CarND-Path/build
Scanning dependencies of target path_planning
[ 33%] Building CXX object CMakeFiles/path_planning.dir/main.cpp.o
[ 66%] Building CXX object CMakeFiles/path_planning.dir/planning.cpp.o
[100%] Linking CXX executable path_planning
ngt@ngt-limited ~/Adipster/Student Reviews/August 2017/27/275. 699959-26-08-2017-pp-ken/jwdunn1-CarND-Path-Planning-Project-509e4f2/build
Map file: ./data/highway_map.csv Driving style is Bold
Listening to port 4567
Connected!!!
```

```
Lap 1
t: 0.0 s: 124.834 Launching...
t: 1.06 s: 129.284 Switching to adapt mode
t: 1.22 s: 130.9 Advancing cautiously
t: 1.22 s: 130.9 Cruising
t: 11.27 s: 355.963 Changing to deeper lane 0
t: 26.09 s: 676.976 Adapting to traffic ahead
t: 41.1 s: 925.85 Advancing cautiously
t: 62.15 s: 1285.53 Changing to lane 1
```

