UDACITY

Back to Self-Driving Car Engineer

Kidnapped Vehicle

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear Learner,
This was a brilliant submission and I enjoyed reviewing your work. By carefully going through the project, it shows a lot of effort, diligence and above all, understanding of the project. Excellent work! 👏👏 You have successfully passed all the rubrics of this project. All the efforts you put in to complete the project are very much appreciated and it was my pleasure reviewing this wonderfully implemented project. You should be proud of yourself because success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do. Please keep practicing on these projects and I wish you all the best.💪

To be honest, I really like the way this project was implemented. Please keep up the hard work!👍

> **We must endorse the hard work and determination we perceived in this implementation and want this spirit to be maintained through out learning with us and we hope you will soon become one of us. Please keep up the spirit!👍**

`In case you encountered some difficulties while implementing this project, please, feel free to share them.` **Thanks!!!!**

Accuracy

This criteria is checked automatically when you do ./run.sh in the terminal. If the output says "Success! Your particle filter passed!" then it means you've met this criteria.

Awesome!

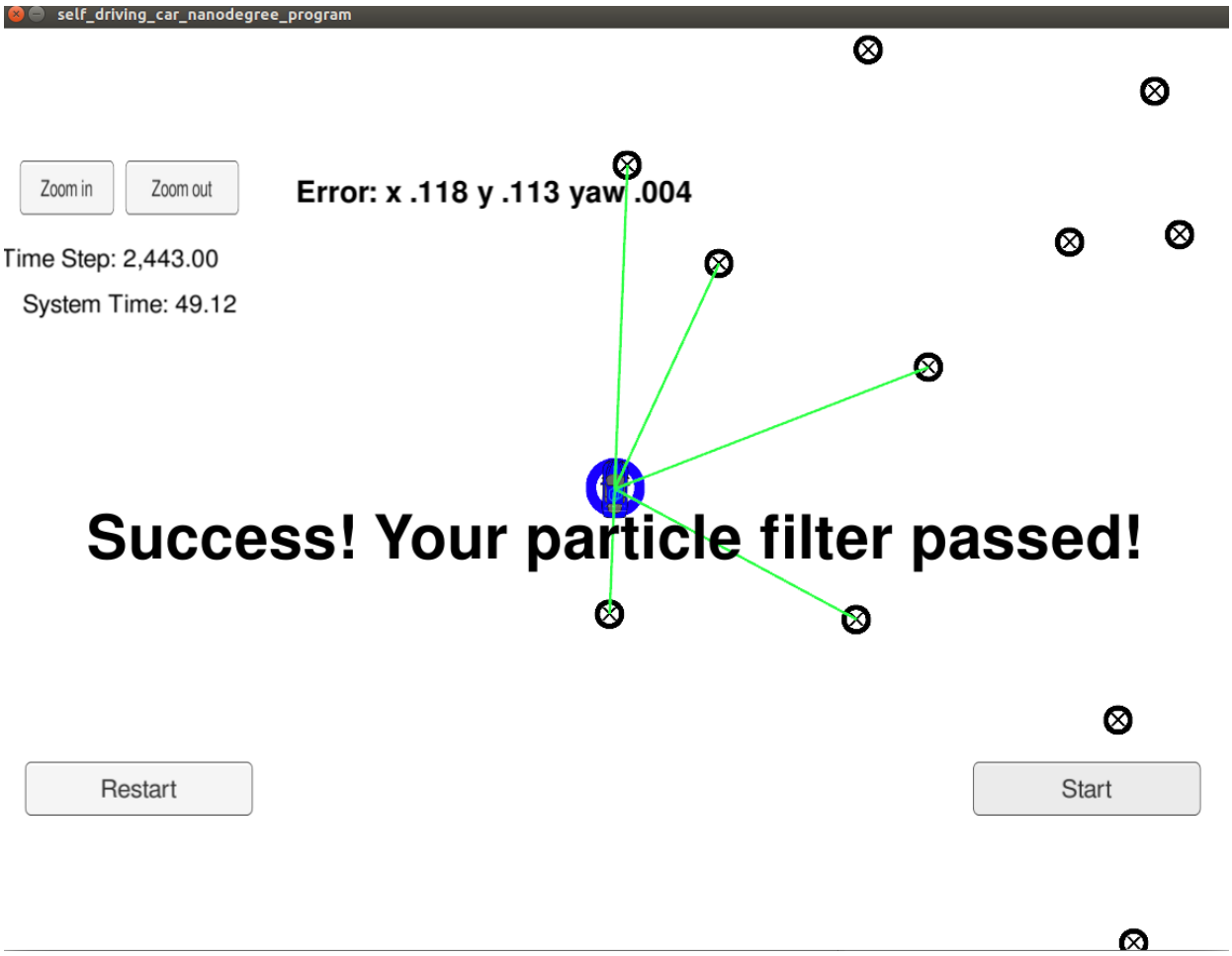
Great job here. Your code can be compiled without errors with `cmake` and `make`.

```
[ 66%] Building CXX object CMakeFiles/particle_filter.dir/src/main.cpp.o
/home/stk/Downloads/jef/Kidnapped Vehicule/1/1/abdullaayyad96-CarND-Kidnapped-Ve
hicle-Project-ab03d9c/src/main.cpp: In lambda function:
/home/stk/Downloads/jef/Kidnapped Vehicule/1/1/abdullaayyad96-CarND-Kidnapped-Ve
hicle-Project-ab03d9c/src/main.cpp:103:32: warning: comparison between signed an
d unsigned integer expressions [-Wsign-compare]
             for(int i = 0; i < x_sense.size(); i++)
                            ^
[100%] Linking CXX executable particle_filter
[100%] Built target particle_filter
```

Performance

This criteria is checked automatically when you do ./run.sh in the terminal. If the output says "Success! Your particle filter passed!" then it means you've met this criteria.

Excellent job!
Your runtime is `49.12` seconds, which is much less than 100 seconds. Nice work!👏



General

There may be ways to "beat" the automatic grader without actually implementing the full particle filter. You will meet this criteria if the methods you write in particle_filter.cpp behave as expected.

The methods in `particle_filter.cpp` behave as expected. The code is clear, logical and well commented. Nice work.👍

Tips

You can study more on random number generation in C++ by refering to:

- Random Number Generation in C++11
- Rand
- C++ TUTORIAL - RANDOM NUMBERS
- How to generate a random number in C++?
- THE C++ RANDOM NUMBER GENERATOR

DOWNLOAD PROJECT

RETURN TO PATH