February 7, 2016 cd096

**Code Review**

The reviewed mapper code was relatively organized and easy to understand but had room for improvement. To improve the code’s style, the team could have used more commenting, removed single function files, and utilized their libraries more efficiently.

There are some files that only implement single functions. These could have been moved to the m1.cpp file or any of the files they created. As seen in Fig. 1. in the appendix, in the DistanceFunctions.cpp file only one function is implemented. Although the use of single function files improves the understandability of that particular function because of the lack of clutter, someone reading the code for the first time may find it difficult to understand the structure as a whole because of the disjointed nature of this style choice. This is important because it could potentially make the code as a whole simpler and easier to understand.

Although the code has comments to describe what each function is supposed to achieve, there are no comments to assist a reader in understanding how the functions work to complete their functionality. In Fig. 2., you can see that although there is a comment to briefly describe the function, the coders have not provided any comments that would help to guide an outside coder through the code and help them understand how the function works. The relevance of this being that, for another coder to edit or improve your code they must first understand how your function works. This process could be made more efficient with more descriptive commenting.

Another possible style improvement would be for the team to utilize the code libraries more efficiently. As seen in Fig. 3., the math.h library has been included in the m1.cpp file. Despite this, in Fig. 4. you can see that the team has written a cosine function even though it is available in the included math library. If the team want to use the function they wrote, another suggestion could be to put the implementation in one of the other files already created instead of directly before the find\_distance\_between\_two\_points function that calls it. This way the flow of the m1 file is not disrupted and it uses the other files more efficiently.

The team also made use of numerous good coding style ideas such as using vertical white spaces and outsourcing some of the functionality to other files. Putting white spaces between different sections of a function, makes the code easier to read and understand. By implementing some of the functionality in other files, the team reduced the crowdedness of the m1 file and improved the readability of the code.

The overall style of the code is adequate. Although some sections of the code are well organized with white spaces, the code style as a whole could have been improved by removing single function files, using only necessary libraries, and making the code less disjointed with comments to guide other coders from one file to another and through different functions.

**Appendix**

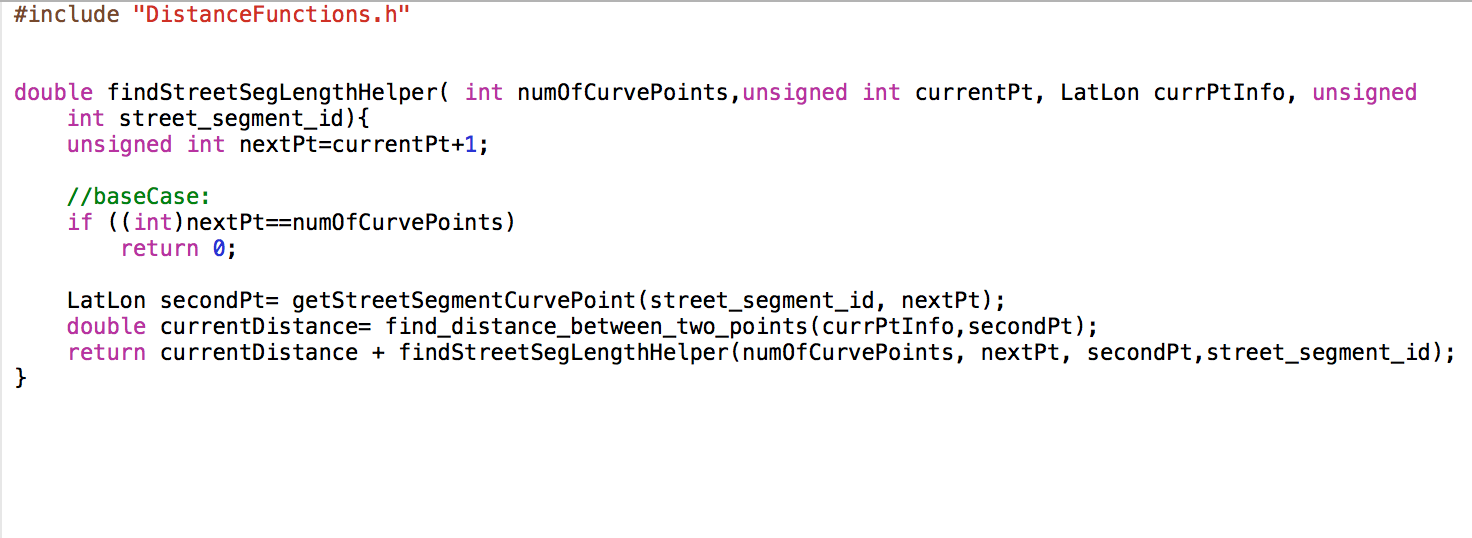
****

Fig. 1. DistanceFunctions.cpp that contains a single funciton

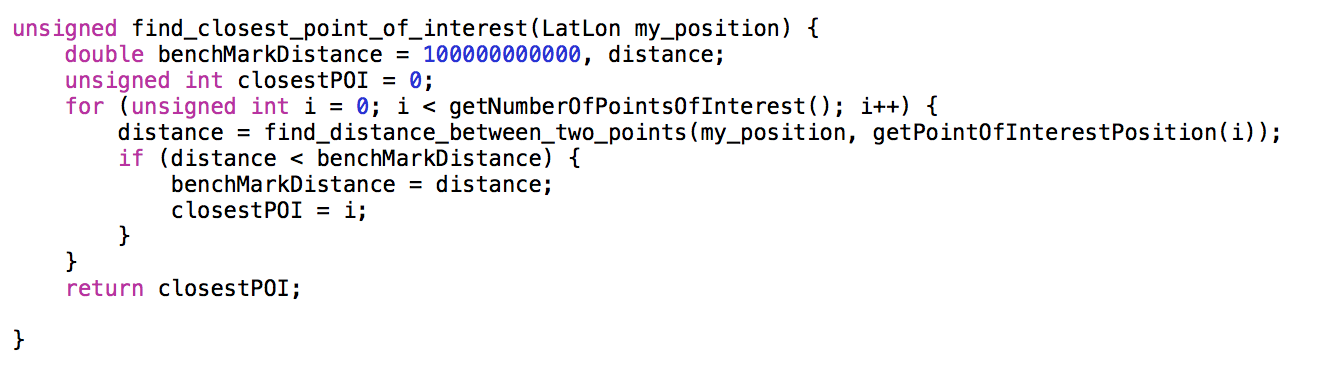


Fig. 2. Example of uncommented function from m1.cpp file



Fig. 3. All the libraries included in the m1.cpp file

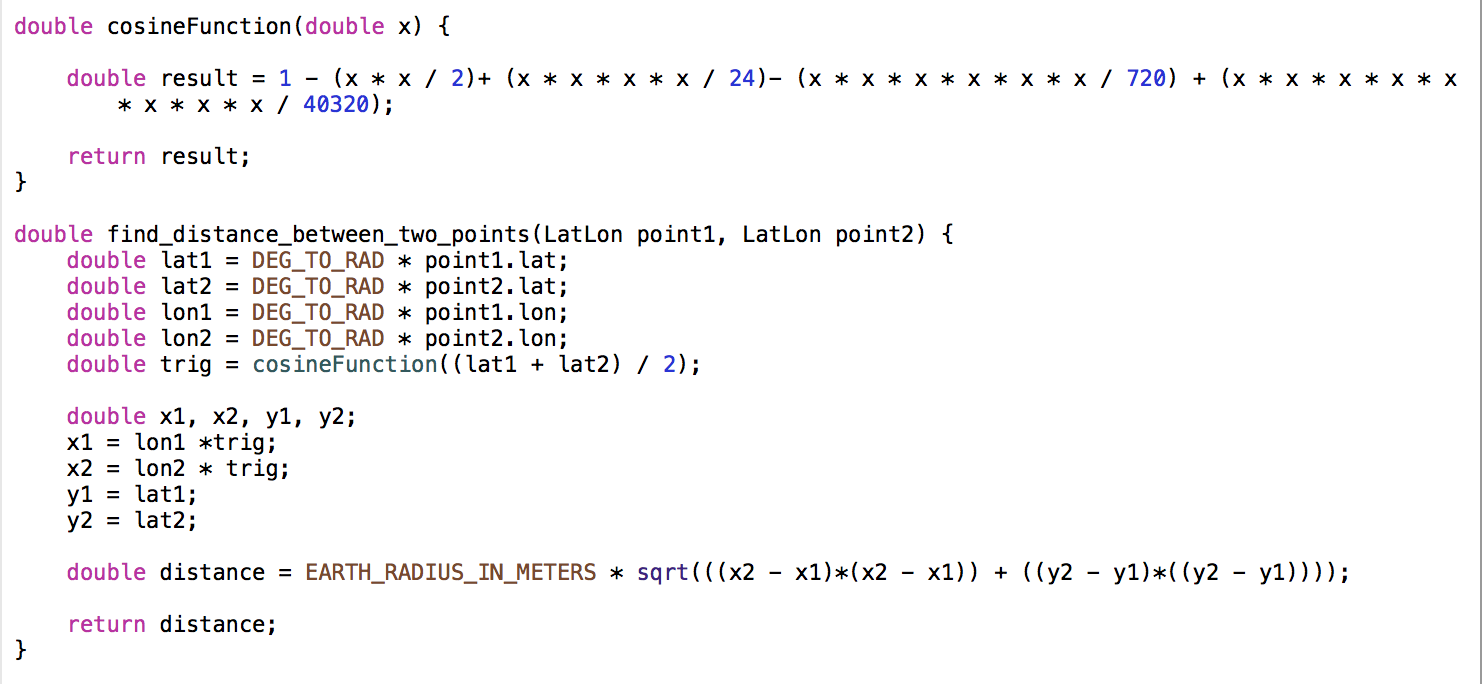


Fig. 4 The cosineFunction and the find\_distance\_between\_two\_points function that calls the cosineFunction in m1.cpp