Assignment 9i, SU

Aditya Fadhillah (hjg708)

Documentation

My task is to analyse the documentation and comment on what is good and bad practice concerning the two proposed documentations.

Fast Inverse Square Root

I will analyse the comments on line 8 and line 9. Comments like this may seem funny, and perhaps there are some inside joke behind it to other fellow programmer, but these comments are unhelpful and therefore unnecessary. But more than that, these kind of joke comments when they are present in a professional code they end up making the programmer that wrote them look unprofessional. The comment on line 12 states that that line is the first iteration, but as it is a loop it will not always will be the first iteration, this comment is misleading. The comment on line 13 is unnecessary as it comments a commented out code. A commented out code needs to be deleted.

Insertion sort

The overall comments of this code is well done, the programmer writes the majority of their comments about the code above it. This prevent visual obstruction when reading the code it self. The comments are also divided into new lines, this makes the code look cleaner and makes it easier to read. The single comment inside the main code, is small and therefore unobstructive. And this comments also helps the reader understand the code better.

Refactoring

Statemachine

We have used dictionary to read the text file for the different level, so we have come to realize just how effective and scaleable dictionary is. The current Statemachine uses switch case to switch between the different states, it can therefore be a bit troublesome to add a new state. But with the use of dictionary it can be easier to add a state, as you can simply add a new state inside a state dictionary.

Propose a refactor

In the first iteration of our ReadTextFile method we have not divided the different functions that separated the different data for (map), (meta), and (legend) into three different string arrays. So our ReadTextFile method was unnecessarily long. We then refactored ReadTextFile by dividing those three functions into three methods, which is then called inside ReadTextFile method. We have not changed anything, we have simply turned the functions into methods. This refactoring makes it easier to read and understand what ReadTextFile method do.