Code Smells João Amorim 57409

1º - Data Class - net.sf.freecol.client.gui.mapviewer.GUIMessage

A screenshot of a computer program

Description automatically generated

This class does not serve much purpose but to represent a message in a chat having tree variables and 3 getters for each not having any other functionality tackling strictly only data.

The way it looks a good idea for refactoring would be to develop this class more give more of a purpose then just contain data maybe there are behaviours that are outside this class, like methods or even variables, that should be moved to this class.

2º - Long Method - net.sf.freecol.client.gui.mapviewer

Method Paint Map

A screen shot of a computer program

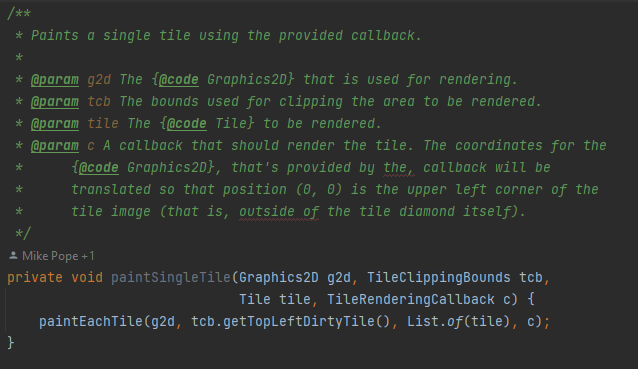
Description automatically generated

This method is just way to long making it way more complex, although it may be a method that tackles the graphical part of the game and its normal for a method with that job to be extensive and complex, the way this method looks with about 174 lines of code and to add insult to injury is not documented make it a smell.

My suggestion for a refactor would be to make it simpler and to better document the method itself and each step done in it, this should not be too arduous because there already some comments in the method that document the step being done. Now when I mean make it simpler this would be the steps that I mentioned before each should be turned into their own method that is called inside the parent method and as I said already all these changes should be carefully documented having in mind the complexity of the method that we are looking at.

3º - Duplicated Code- net.sf.freecol.client.gui.mapviewer

method paintSingleTile



Although its well documented and the purpose of it is well understood, this method is not used anywhere and there is even alternative to it in the same class, so with what was just mentioned this method is useless there are no comments justifying the “why” it’s not used or even if there is an “when” is going to be used making redundant and just unnecessary complexity to the code base.

A good way to refactor this would be firstly to understand if there is actually a use for it, and this can be achieved but exploring the code base or even making a pull request with the changes and seeing the opinions of other collaborators, if there isn’t an use I would just remove it.