Exercise 2 - Software Metrics

1. The analysis file can be found in the written deliverables folder, under folder Iteration IV.

2. a) The biggest design flaw concerns the TextureBox class, with a total of six flaws. Three of those are complexity flaws and the other three encapsulation flaws. All the flaws revolve around the constructors, which each having a long list of parameters. This causes the class to be affected by Data Clumps.

   The next flawed class is a test class, the CollisionDetectionTest. Three cases of internal duplication was found here. Three test cases uses nearly the exact same code for each case. Since this is a test class that does not affect the workings of the code, we decided to ignore these flaws.

   The following flawed class is the Box class. Just like the TextureBox class, the constructor of this class has a long list of parameters. This class is thus also affected by Data Clumps, which affects both the complexity and encapsulation.

   The last most flawed class is the Mouse class, which has Feature Envy flaw in one of the methods.This design flaw is caused by a method which determines the Mouse position. This method uses a lot of variables from Launcher, which is the reason of the design flaw.

   Other design flaws include Data Class design flaws, which there are four cases of, and Data Clumps design flaws with five cases.
   Both the Circle class and the ScorePopUp class are noted as Data Classes. They have several methods that don't get used or only get used privately, exposing data via public accessor methods.

   The other two classes affected by Data Class design flaw are the Launcher class and the GameVariables class, both having a good amount of public accessor methods. The classes affected by Data Clumps have, as earlier described, long parameter list. These are all objects that use position coordinates.

   b) To fix the the Data Clumps for the TextureBox class and Box class and all other classes affected, the decision was made to replace parameters that were always grouped, such as position coordinates float x and float y, with a Point object that contains the coordinates. The parameter list is now shorter for all objects that needs position coordinates and all seven Data Clumps design flaws have been fixed.

   Some cases of Data Classes have been fixed by removing some unused methods or methods that were only used privately, therefore reducing the data that is available for public. The fixed classes include the Circle class and the scorePopUp class.

The feature envy flaw of Mouse class has been fixed by splitting the class up in two classes (MouseButtons and CursorPos). The feature envy design flaw was caused by a method used to transform a Pixel position to the corresponding OpenGL position. This flaw was solved by moving this method to the Launcher class.

With all these design flaws fixed, this leaves only two design flaws left in the code. The GameVariables class is still noted having a DataClass design flaw. While we're willing to fix it, we decided not to. Its function is to hold all attributes and data of the game and therefore many public accessor methods are needed to acquire these data. To overhaul this class we would have to split all the data to various classes. Therefore we decided to ignore these flaws.

The other flawed design left is the CollisionDetectionTest with its duplicated code. This doesn't affect the inner workings of the code and changing it would cause more difficulty with testing this class. This flaw has also been ignored.