Assignment 4 2020/01/24

Exercise 2:

• Register and Login classes:

Before:



Register and Logic class both make a use of displaying notifications when data is invalid. This allowed us to apply extract method metric, which reduced Lack of Cohesion from "medium-high" to "low-medium" once method generateNotification() was moved to ValidatorService class.

GameOver class:

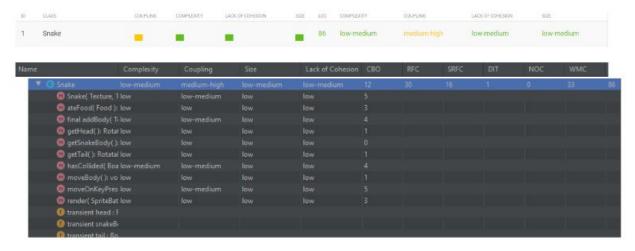
Before:



This was not the biggest improvement since only the size of methods ware reduced to increace cohesion, but the reason for that is that the GUI classes rely on object like Labels, Buttons, Tables, Stages in order to display all these objects in our game. It was possible to extract logic components from the constructor method and make it more granular, thus increasing cohesion of the class.

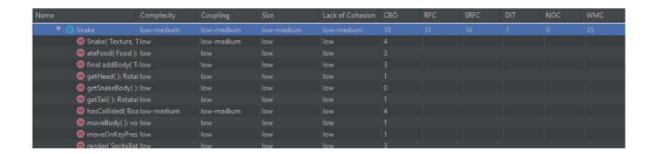
Snake class:

Before:



After:





We chose to focus on the CBO metric as improving this metric in specific methods will also improve the class score. The CBO metric means the 'Coupling between object classes'. This metric applies both to methods and classes, and by refactoring the methods the class' score also improved. For the Snake class refactoring, we used the Extract class technique, by introducing the new class 'Movement'.

UserDao, ScoreDao, MySqlConnector classes:

Before:

List of all classes (#3)										
ID:	CLASS	COUPLING	COMPLEXITY	LACK OF COHESION	SIZE	1.00	COMPLEXITY	COUPLING	LACK OF COHESION	SIZE
1	UserDao		•	•		116	low-medium	low	low	low-medium
2	ScoreDao		-			84	low-medium	low	low	low-medium
3	MySqlConnector			-	-	11	low	low	low	low

After:

LeaderboardDao

4 MySqlConnector

11

low

low

low

We extracted method to get top 10 scores from the database in ScoreDao class to the new class called LeaderboardDao. This provides more logical separation of concerns, reduces complexity, class sizes and lines of code in the classes.