

Assignment 3 Code Review Report

The purpose of this report is to outline the issues found in the codebase of a game application and the steps taken to address them. The application in question is a game where the player navigates a maze, collects objects and evades enemies. The codebase was reviewed and several code smells were identified, which were addressed through refactoring and documentation.

Firstly, we noticed that the `pickUpObject` method in the `Player` class has many duplicated code lines for performing the same functionalities. To address this issue, we refactored the code by modifying the repeated parts into small methods and reusing them, resulting in a cleaner and more organized code. the updated part could be found on this link ([e3dd78ae](#))

Secondly, we found an unjustified use of primitives in the `Enemy` class. One of the variables called `"takeNextX"` is defined as a Boolean object wrapper. We updated this variable to be defined as a boolean, which is a primitive type. This updated could be found in ([e3dd78ae](#))

Thirdly, we identified an unused or useless variable in the `setNode()` method, and we removed it to simplify the code. ([bbd18b22](#))

Fourthly, we noticed a lack of documentation in some parts of the codebase. To address this issue, we added some documentation to improve code readability and maintainability. And this changing is in ([fdad87f9](#))

Fifthly, we found an unnecessary use of unsafe or unsound constructs in the `FindPath` class. Specifically, in the for loop in the `resetNode()` method, it wrote `'int j = 0;'` then inside the for loop it wrote `'j = 0,'` which is a repeat problem. We updated it to `'for(int j = 0;...)'` and removed that code. This particular revised part is in ([e3dd78ae](#))

Sixthly, we identified a long list of method parameters in the `pickupObjectEffect` method of the `Player` class. This could cause confusion when reading the code. We refactored the method structure and removed one of the parameters to make it more organized and easier to read. Here is the updated code ([8f4576d4](#))

Finally, we also renamed some class names to be more understandable. For example, we changed `'Obj_Cheese'` to `'Cheese.'`. It can be seen in here ([8f4576d4](#))

In conclusion, our code review has identified several areas for improvement, including code duplication, the unjustified use of primitives, unused variables, lack of documentation, unnecessary use of unsafe or unsound constructs, long parameter lists, and unclear class names. By addressing these issues, we can improve the quality of the codebase, enhance maintainability, and ensure that the code is more readable and understandable.