Report

Zhifang Wang
the University of Auckland
Auckland, New Zealand
zwan686@aucklanduni.ac.nz

I. The strategy of naming

In this assignment, for most fields, variables, methods, and parameters, when I named them, I follow the principle that all names have real meaning and what they represent. I did not use a single letter to name them. It is mean that if other developers reviewed my code, they would understand my code quickly.

As mentioned in the lecture, I named them almost entirely in full English to make my code more comprehensible and maintainable. Still, for some variables, I also use abbreviations, like ‘num’, because abbreviations and full English identifiers have the same effect in some cases. I also use Camel-Case to improve the readability of my variable names

II. Encapsulation of methods

In this assignment, I have wrapped some of the methods. And gave them meaningful names. This has improved the readability of my code and improved comprehensibility for future maintenance.

If I need to refactor these methods at a later stage, I can quickly find a method and reprogram it, which also improves the changeability of the project.

The encapsulation of methods also improves the overall reusability of the code. This also dramatically reduces the amount of repetitive code that occurs.

In some methods, I use privately for modification instead of public, reflecting the object-oriented feature of security.

I didn't choose to complete the logic of the entire game in one method because there was so much code that was repeated. And doing so would make my code very difficult to understand. When other developers see my code, they're confused, which is terrible for later maintenance.

III. Comment

Next to some methods, I have annotated them in detail, explaining the method's function.

Also, when it comes to game logic, I have described each case in detail, such as suitably running this method. This allows me and even other developers to know the meaning and logic of my code.

Such comments improve the readability of the code, as well as its comprehensibility.