Team T-04
Tse Ching (Maisy) 20598883

COMP3111 Project - Supplementary Note

Change of algorithm for task 6

In task 6, the compatible score oScore is said to be "Provide a score of compatibility (0%: Not Compatible; 100%: Perfect Match)". However, in the provided NK-T6 Algorithm, oScore is calculated as $(1 - \text{abs(oRank - oRankMate)} / \text{oRank}) \times 100\,\%$, which may produce a negative value. For example, if oRank = 2 and oRankMate = 20, $\text{oScore} = (1 - 18/2) \times 100\,\% = -800\,\%$.

I believe a score in the range [0%,100%] would be more "intuitive" as

1. It is presented in percentage, where most people will expected it to be a value between 0% and 100%.
2. It is more easy to compare the difference of multiple score as there will be no negative value and we know what is the exact range.

Therefore I introduce a new algorithm for calculating the score.
Let $\text{diff} = \text{abs(oRank} - \text{oRankMate)}$ (i.e. difference between the ranks of two names) and
$\text{sum} = \text{oRank} + \text{oRankMate}$.

$$\text{oScore} = \left[1 - \frac{\text{diff}}{(\text{sum} + 200 + \text{diff})}\right] \times 100\,\%$$

Explanation:
I'd like to produce higher score (not below 40%) so that users will be happy.
For $\frac{\text{diff}}{(\text{sum} + 200 + \text{diff})}$, it is the factor showing the difference between user name's rank and soulmate name 's rank. As $\text{diff} < \text{sum} + 200 + \text{diff}$, the factor will be smaller than 1 (and should be not close to 1). Then, 1 - factor would be not close to zero and thus a higher score.

In the equation for calculating the factor, the "+ 200" is used to make factor smaller when sum is very small while the "+ diff" is used to make factor smaller when diff is large.

6 separate classes for computing the tasks

Our group has 6 separate classes (not inherit a common class nor having a common interface) for computing the result for the 6 tasks and here is the reason. In the second meeting, we found out that we are likely to have different "ways" to implement those 6 classes and they don't have much similarity. Later I found out my two task seems to have some common structure and it turn out we may able to have a super class for them or make them to implement a common interface. However, we have finish most work by then. And again, three of us are have different "ways" to implement those 6 classes. So we didn't change this structure. Next time I might need to have better planning.

Package structure

At first, we think that we won't have much class for the project so we didn't plan to have sub-package. Later I found out it is a bit "messy" as three of us are having different naming style (e.g. different prefix for those 6 task computing class) and all type of class (ui-related class, utility class, the 6 computing classes) are "mixed" together under the same package. It will be much better if we divided them into different sub-package. But again, as most code are done already and I am afraid I would make the other's code to fail, I didn't make this change at last.

Presentation of output

I found out it is a bit difficult to come up with a stable and tidy text table to show the output in the console. I have thought about adding a table in the UI and put the result in it but it will affect my teammates' work and I don't really the document / example about it I found online (sad) so I give it up. I then decided to change the front of the console to a monospace one. This help a lots.

Things that I need to improved & something I learn from the project

- It is always difficult to name the class, function, or even writing comment and git message. Sometimes, I found out I am having a bad naming / message but I can't change it or I have to change it in multiple places.
- Some of the class I created have some private method. It will be more clear and easy if I can test them separately but I can't. So I am not sure about how to write test case so that all of them would be welly covered.
- Need to have better planing about similar class and the package structure next time.
- (By the way, it seems that my class for computing the task result are pretty long. I am not sure if I am making the simple problem too complicated or what. But I believed dividing the functionality of the class into different separated method will be more easy to maintain / follow.)