

# Assignment 4

*Rui Qiu #999292509*

## **Description.**

Our group is working on the command-line interfaces and trying to add some new features to our prototype to fulfill some extra user needs. Our initial solution to this is to redesign a new command-line interfaces other than the current mainstream ones.

First I have to claim that in order to make auto-correction and auto-complete features possible, we need to construct a valid commands library (call it VCL). And this library is dynamic because it will be modified and updated based on user's preferences from time to time. For example, if a user always inputs command B after command A, then command B would be the first suggested when this user inputs command A again. And by default the top three suggested auto-complete candidates will appear on screen, the user can choose the desired one by pressing number key or continuing typing other commands (which is not the candidates).

For the general interface design, I would like to divide the command-line interface into two columns: one is for normal usage, the other is for status display. Specifically:

- when an issued command is logically correct (according to VCL), a check mark will appear in the corresponding line in the right column.
- when a command is problematic (before hitting 'Enter'), which is not in VCL, a question mark will appear on the right hand side.
- when a command leads to an error, a warning sign will appear correspondingly, and at the same time, the error message will appear on right instead of one line under the previous command.

Last but not least, the anti-fatigue feature is designed to be a pop-up window, which appears at the center of the screen every one hour.

All of the features mentioned above should be enabled/disabled in system preference. And for more details, please refer to pdf file.

## **One future scenario of use.**

The forgetful Jason always forgets how to issue some certain commands in command-line interface, for example, he does not remember which one is for log information, '-m' or 'm-', so that he has to turn to Stack Overflow or Google for help. But with this newly designed interface, when he typed something wrong with that command, a curly line shows up right under the incorrect command and the suggested correct command is shown in a highlighted box next to the command. Therefore all he needs to do is just simply click number 1 to select that correct command. Hurrah!

**Critical assessment.**

Since the user needs in our Phase III covers more features, therefore my design is quite limited in some ways. For instance, the interface design itself is very basic and simple, some other elements should be included as well in order to fulfill features left. The design is not beginner-friendly enough. In some situation, a beginner user of command-line interface would still feel confused about what command to use.

However, it does offer a solution to the following needs:

- Auto-correction / auto-complete of arguments and commands;
- Output message after successful command;
- Error messages should not be obtuse and should be helpful in all cases;
- Eye-fatigue-proof.