

The description about the Undo-Redo designs

It can be beneficial to practice the command and memento behavior design patterns many times. The more I have practiced, the more details I have found.

The exam coding project is quite different from the project of Assignment 03 and Assignment 04. The exam project contains more commands while the assignment project is mainly about add command. The advantages of the Memento Design Pattern are more pronounced since I have created five different command classes for Command Design Pattern in Part I and the project eventually became bloated. Apart from the numerous command classes, another flaw in Command Design Pattern is that it is needed to maintain the undo method for each command, while the Memento Design Pattern simply achieves the undo and redo by resetting the state that has been stored in advance.

However, it does not mean that the Memento Design Pattern is a silver bullet. Compare to the command pattern, it consumes more memory as it stores the whole state every time the state has been changed even the little change, like only moving up 1px for the mouth! Mover over it is hard to see the details about the command when executing the redo or undo function since the memento pattern primarily focuses on the state instead of the command. This can be easily achieved in command pattern, simply set the detail in do and undo method in the command class.

To sum up, there is always a trade-off for between the two design patterns. In this case, the memento pattern worked better for my solution. As it is a ticklish business to maintain so many commands. More importantly, the memory consumption might be not that serious since the state of the solution is just the Face class that only contains five properties which are two brows, two eyes, and one mouth.