# Command Vs Memento

Both the Memento and the Command pattern were highly effective in implementing an undo and redo system in my SVG code generator.

Although, I found that the Command Design Pattern was the one that worked easier for my solution. Overall, I found it to be the easier to implement as well as understand, whereas with the memento pattern I struggled to follow what code is sent to and from classes and how it effects the code generated. I enjoyed the straightforward nature of the command patterns ability to store all the information to perform an action inside an object. I found the pattern to be more of a defensive mechanism in this case, and it proved to be more reliable within the code. It is a pattern which I would prefer to use over memento in a future project, as I found the memento to be quite complex.

I had struggled a lot initially with implementing the memento pattern. I put this down to attempting to pass a list of strings through the originator and caretaker classes when adding, removing or undoing a shape as required. Eventually I downscaled my design to work with one singular string as opposed to a list, and create line breaks as the string gets increasingly large, to ensure that the output is not reading as one big line, in which a scalable vector graphics generator may reject. In doing this I was able to get a better understanding of how the code works and what each class is responsible for.

To add to this, I don't believe that either design pattern would get more complex than the other as the implementation scales larger. While attempting the extra credit for Assignment 3 using the command design pattern, where I had to add a text class the act in a similar fashion to a shape, I found that this was easy to implement with the rest of my code. I would assume that this would be the similar with the memento pattern, wherein the actual extra work is just to add the new class.

To conclude, I really enjoyed the process of learning about these two Behavioral patterns. I feel it has contributed largely to my understanding of Software Design and has fueled my interest to learn about other patterns.