I found the Memento Design Pattern was a lot less work to implement, once you set it up there is no additional work needed to facilitate more commands. A snapshot of the canvas is simply taken every time the state is changed, and we store those snapshots. Then to go back or forward we simply traverse the snapshots, simple. However, I feel the Command Design Patterns true strength isn’t useful for this type of application, it allowed me to implement undo redo but that’s all. No other reason to encapsulate commands in objects. Them being in Objects allows us to do a lot more with them.

While using the Command Pattern I found it a little tedious having to write a new class every time I wanted to add another function that I needed to have undo redo capabilities. So, to answer the question how would the patterns scale if I wanted to increase the amount of commands, horribly. Say we had an application with 100 commands, I shudder at the thought of having to write 100 classes each with their own 100 executes and 100 unexecutes. And while executing a command may be straight forward there is no general formula to unexecute it, every situation needs to be approached differently. This took me some time to understand.

As for the question how, the patterns would fare if the canvas gets larger. If I was expecting a lot of shapes to be on my canvas or even multiple canvases, then the Memento Pattern becomes far less enticing. The extra work in the beginning to implement the Command Pattern would probably pay off in better performance. Due to both less steps being carried out by the computer in order to maintain the canvas states and then of course less memory usage.

I also just want to make a quick note on the Command DP, I found working with it you need much stricter control over the Canvas object as it needs to be statically referenced from each of the Concrete Command classes. So you need to set up some type of global assessor for it.

So, in relation the personal challenges I faced with the Patterns. I found the Memento easier to understand. Easier to get a concreate idea of how it worked. There was more of a learning curve to the Command. But once I got them, they were both okay to work with. Well I say that, but I never did get the Memento working but its because the Memento Objects were storing references to the Canvas. Which is fine but my Canvas wasn’t really a Canvas rather just some methods useful for doing work on the shapes. For instance, my Canvas Class didn’t even contain a list of shapes only a string containing the SVG of shapes. The list was maintained in the Program. It took me a while to understand why this was wrong. But that’s beside the point. I did get the Memento to work in other small projects.