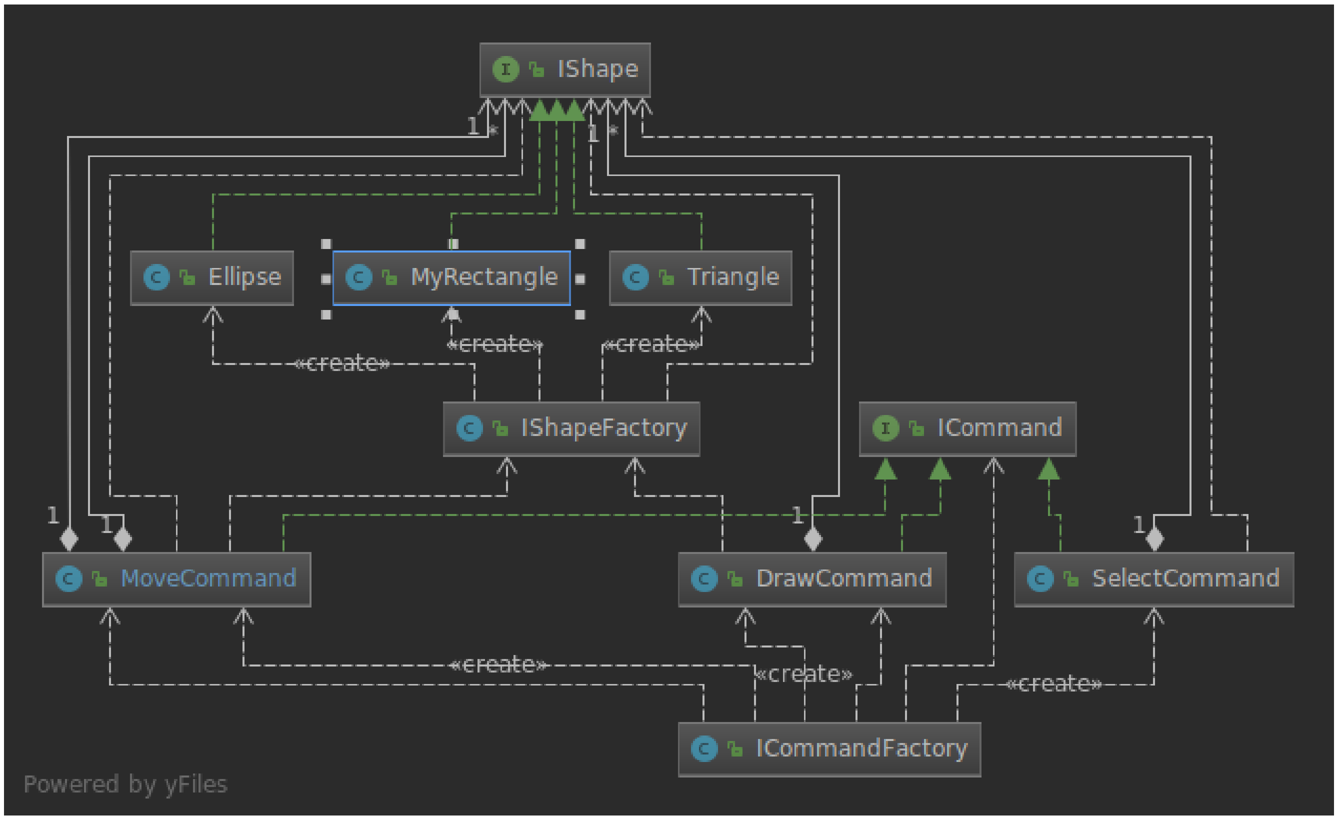
Cory Miljour
SE 450 – Summer 2018
JPaint Final Project

Notes: Program created and tested on Arch Linux and Manjaro Linux.

List of Features: All the features have been implemented except for undo on successive moves. For example, undo works correctly if you move and object(s) once. However, undo doesn't work correctly if you move an object(s), move again, and move one more time. It will bring the shape(s) back to pre 1st move instead of pre 3rd move.

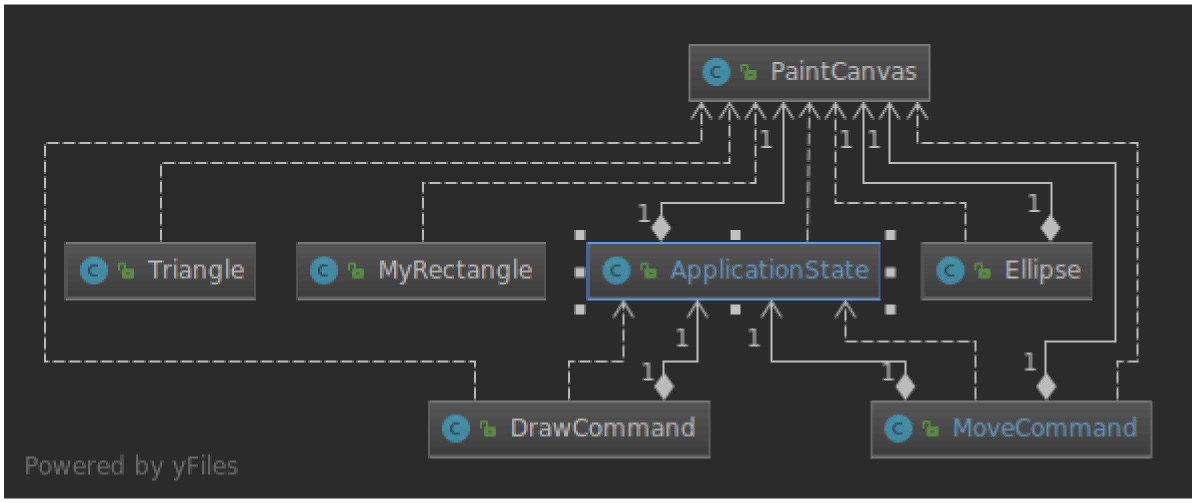
Notes on design:



Design patterns 1 – 2: Factory patterns IShapeFactory and ICommandFactory

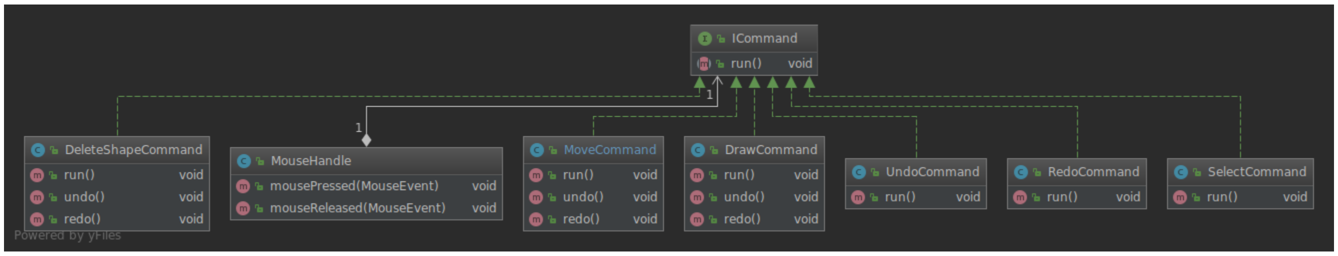
Problems solved: This removed switch statements to allow deferment of instantiation. It allowed the factory to create an object instead of cluttering up code with switch/if statements all over the place. The code is A LOT easier to read after applying this pattern. As you can see from the diagram, the 3 types of shapes are generated from the

IShapeFactory and the 3 main JPaint ActiveStartAndEndPoint modes are generated by the ICommandFactory.



Design pattern 3: Singleton

Problems Solved: This pattern ensures there was only one instance of the canvas. That means all operations will be applied to this one instance. It will stop creation of additional instances which potentially would cause nothing to be drawn to the screen if the wrong canvas instance was targeted. As you can see from the diagram, since multiple drawing classes use the canvas, it is better to make only one instance.



Design pattern 4: Command

Problems solved: This allowed one command "run()" to be executed in an abstract manner for mainly drawing shapes. This one run() command could be used with all shape objects when ready to draw. As you can see from the diagram above, it is heavily used for most operations in relation to drawing shapes on the canvas. If there wasn't one run command, you would have to reference potentially a different method invoker for every shape and misc. other commands. This leads to confusion of code and potential may bugs

Successes and Failures

Successes: I finished with 99% of the features implemented. Learning Java and OOP! I was pleasantly surprised to discover more design patterns while refactoring. I appreciate the time and effort required for this project to enhance all the learning so far in the degree.

Failures:

If you resize the window after JPaint is loaded, the shapes disappear on the screen. If you select anywhere on the screen, and choose "DELETE" for example, the shapes will reappear.

As the design notes state, I could not properly make undo work when you move shapes multiple times. I simply ran out of time and headspace to make this happen.

This project was a slog. Nothing worked immediately. It was a constant create, test, modify pattern of programming. I believe the shapes and the geometry involved were the most difficult part of the assignment. I had to spend a lot of time figuring out x,y coordinates in relation to moving, pasting, and general drawing. The bounding box was difficult to implement completely. It took a while to figure out why the bounding box didn't follow the shapes around. I realized I had to create a new bounding box rectangle with every move and paste. In retrospect, I should have figured out the design patterns as I was writing the code. I spent more time on trying to make the code work than thinking about incorporating design patterns.