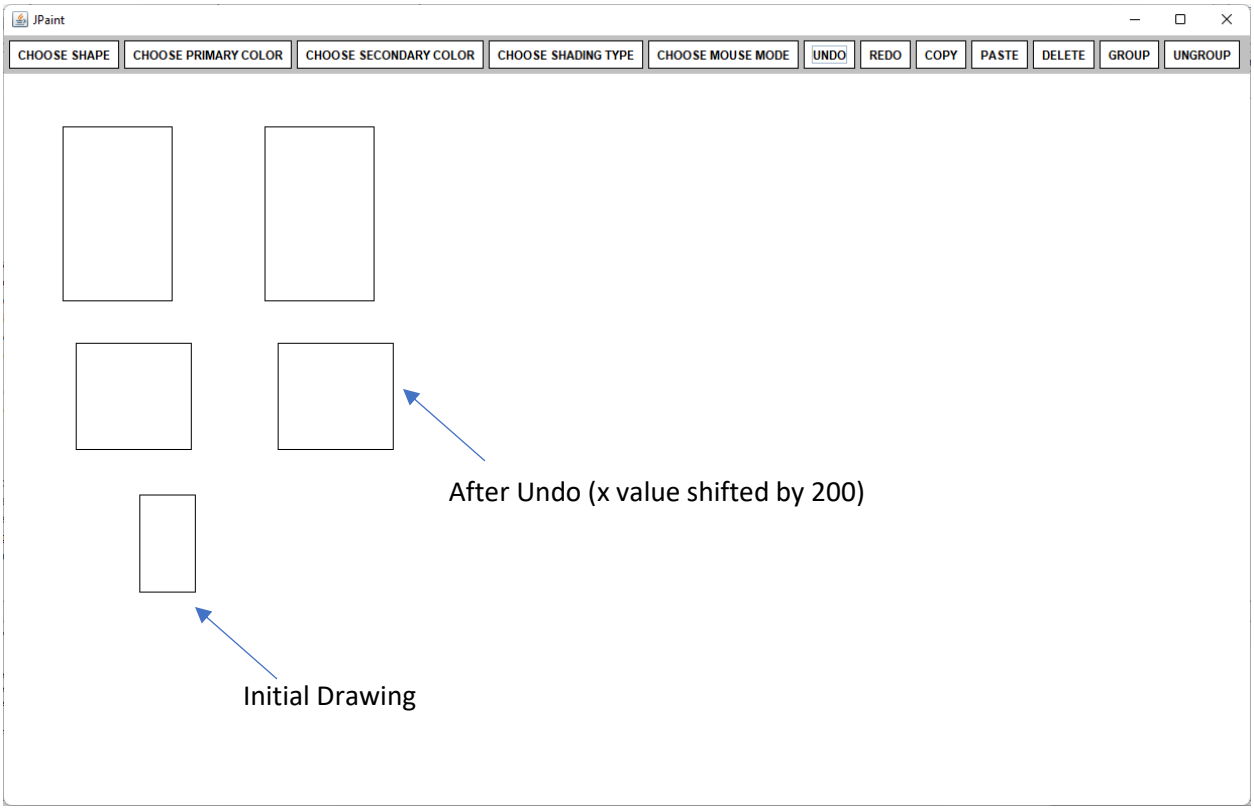
README

Project 1st Check-in

1. **features, bugs, extra credit, and miscellaneous notes.**

I couldn't implement the Undo/Redo correctly due to an issue with repaint() method. It clears out the graphics and when I redraw by retrieving the saved objects inside the stack it doesn't show the new draw. So, I commented out where I called the repaint method. Therefore, old rectangles are still remain on the canvas. To show that undo works I shifted starting "x" point by 200 pixels when I hit undo.



2. **Link to GitHub repo -** https://github.com/sanjayadpf/OOPFinalProject.git

Note: I have given the access. Please accept the invitation.

3. **List of Design Patterns**

Tried to implement Command pattern for Undo/Redo using the CommandHistory.java. Couldn't succeed because of the issue that I faced with repaint() method. Trying to figure out How to use the CommandHistory.java file with the actionlistener of the Buttons. Created a GenerateShape Class which implements ICommand and IUndoable. In there I'm adding the GenerateShape objects to the CommandHistory.

Project 2nd Check-in

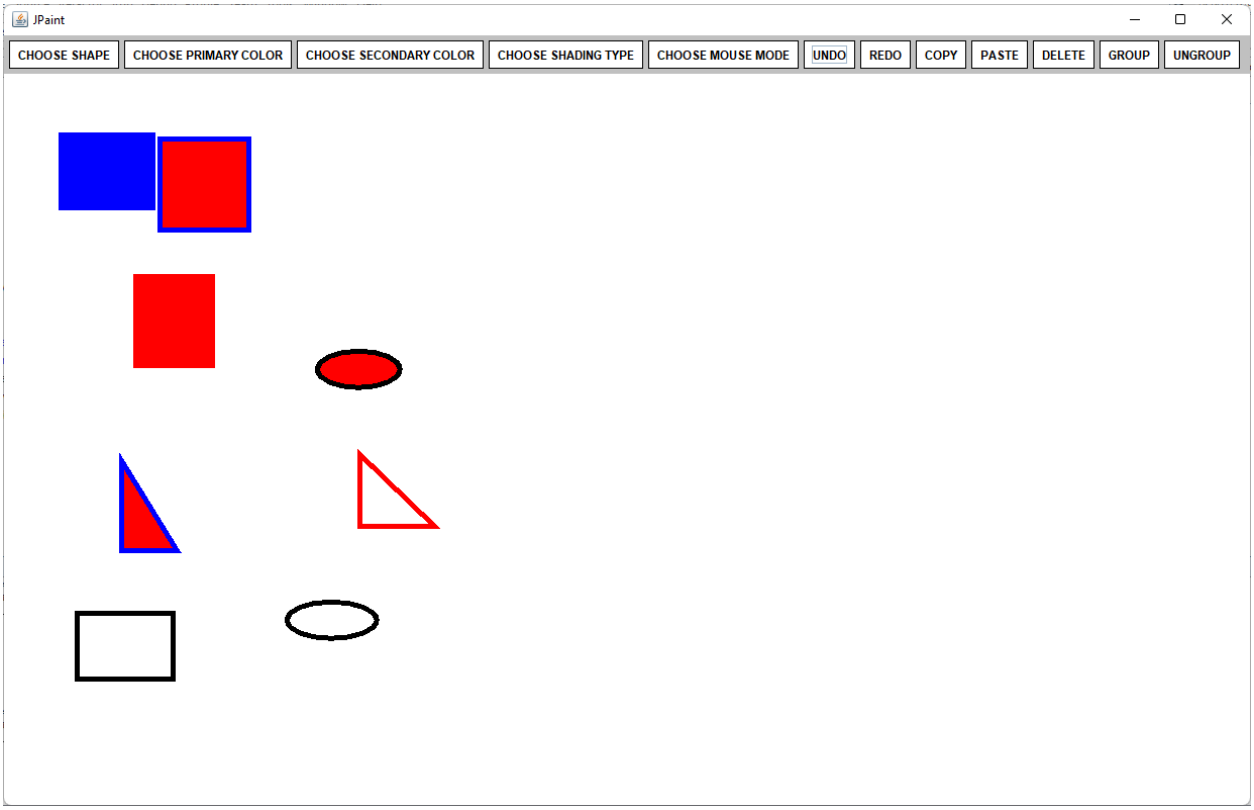
1. **features, bugs, extra credit, and miscellaneous notes.**

I have implemented the button functions CHOOSE SHAPE, CHOOSE PRIMARY COLOR, CHOOSE SECONDARY COLOR, CHOOSE SHADING TYPE. Couldn't implement the functions of mouse mode. So, the entire app is always on Draw mode. I struggled with selecting the shape. So, I uploaded the version which has no selection or move features. Major improvement is I managed to implement the Undo/Redo which I couldn't implement due to the issue I had with repaint() method. Anyway I tried fixing the issue with repaint() method as you suggested in the feedback last time. But I had no luck going forward with it. Instead, I used the setColor() method of Paint Canvas which I can set the Background color whatever the color I want. So I set it to "white" whenever I want to clear(easiest thing I should've done last check-in). I override the toString method of ColorInfo class. This was helpful for me when I was adding the ColorInfo objects to the JOptionPane with input dialogbox. In order to display the Color, toString() method has to override with the String we want to display. For this I referred following links.

https://stackoverflow.com/questions/27518555/pop-up-swing-window-with-both-option-and-textfield

https://www.codejava.net/java-se/swing/jcombobox-basic-tutorial-and-examples

Sample View of the App



2. **Link to GitHub repo -** https://github.com/sanjayadpf/OOPFinalProject.git

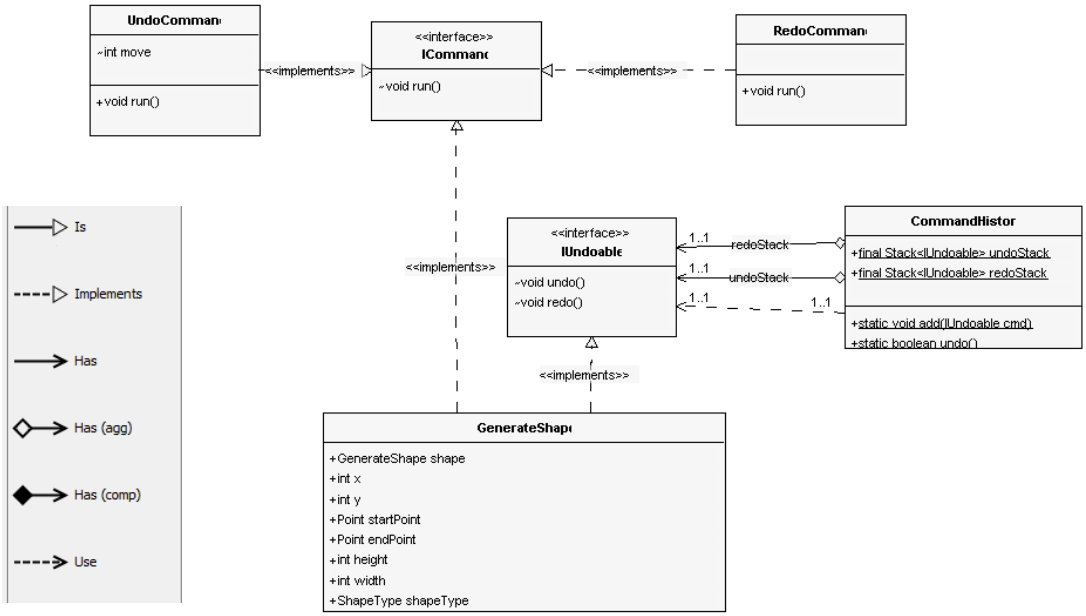
Note: Please contact me if you have any issues with accessing the repo. SMALLIKA@depaul.edu

3. List of Design Patterns

I implemented 3 design patterns discussed in the class up to now.

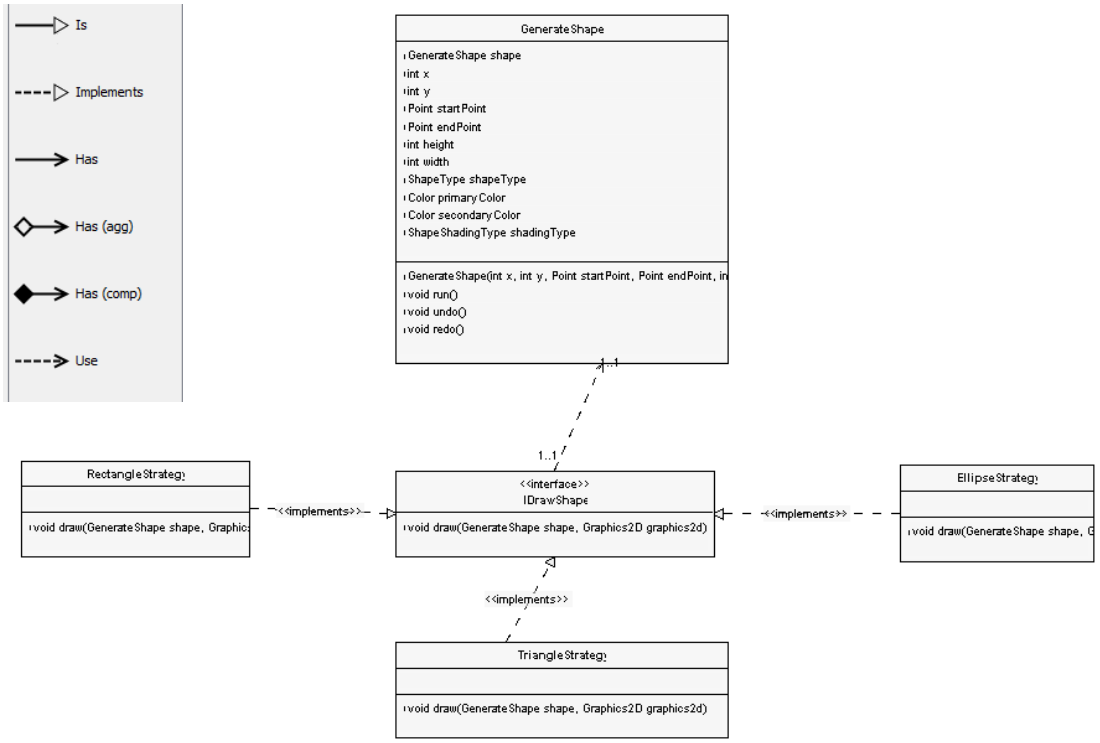
- Command Pattern

This is implemented along with the CommandHistory.java for the UNDO/REDO



- Strategy Pattern

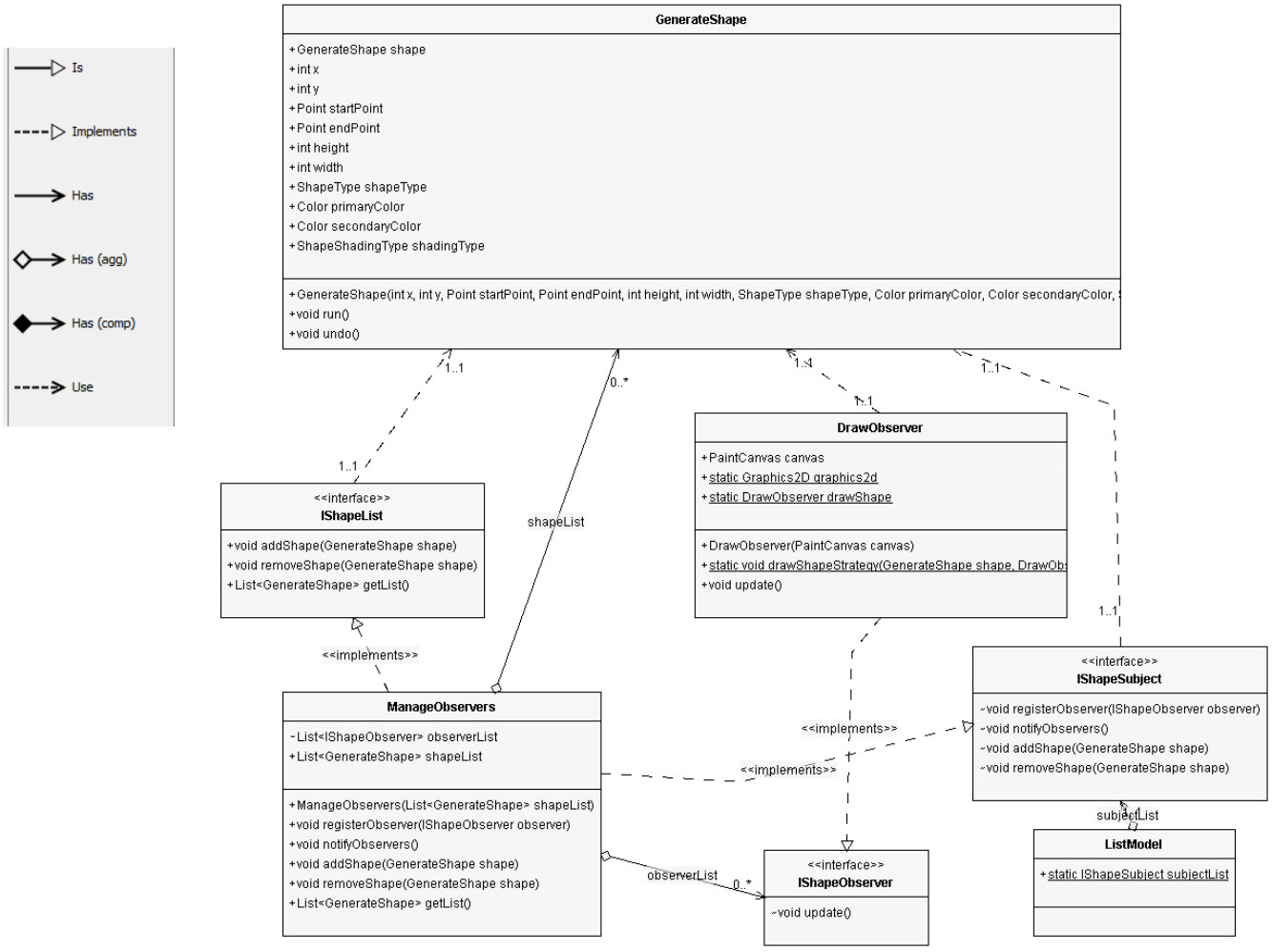
To draw the shape with same draw() method while the object assigned is changing in run time.



Note: Not all relationships are shown. Only the required ones.

- Observer Pattern

To notify the which shape currently is drawn.



Note: Observer is registered in the main method.

Future Development.

- Another Subject/Observer to implement the Mode selection or Static Factory Design.