

## **SE450: JPaint Project Report**

### **Sarun Worrasangasilpa**

JPaint is a Java-base program that works like Paint program. It has basics features like:

- Draw: User can click and when release the mouse. The shape is created according to the chosen shape type (Ellipse, Triangle, Rectangle), chosen color and chosen shading type (outline, fill-in, outline and fill-in).

- Select: User can choose select mode, when click or drag the mouse on/across exist shape(s), the shape will be 'selected' and the outline will appear around the selected shape(s).

- Move: User can choose move mode, when click and drag the mouse, the selected shape will move to the point where the mouse release.

- Copy: User can copy the selected shape into clipboard

- Paste: User can paste the copied shape into canvas

- Delete: User can delete the selected shape

- Undo: User can undo all the previous commands

- Redo: User can redo all the previous commands

\*\*\*In Project overview did not mention to Undo and Redo work on Select function, but I implement the code to make it work (maybe buggy)\*\*\*

### **List of missing features**

- Group: The function that allow user to make selected shapes act like one shape

- Ungroup: The reverse function of Group

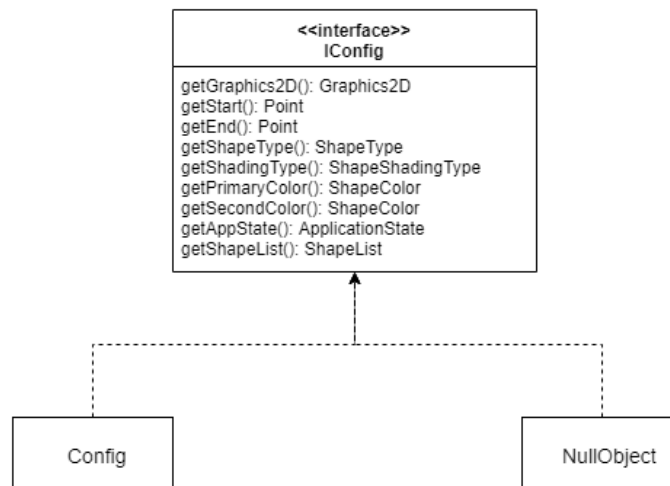
### **Bugs**

- When select the Triangle shape, the outline appear at the rim of shape, while in Ellipse and Rectangle shape, the outline appear around the shape.

## Design Pattern

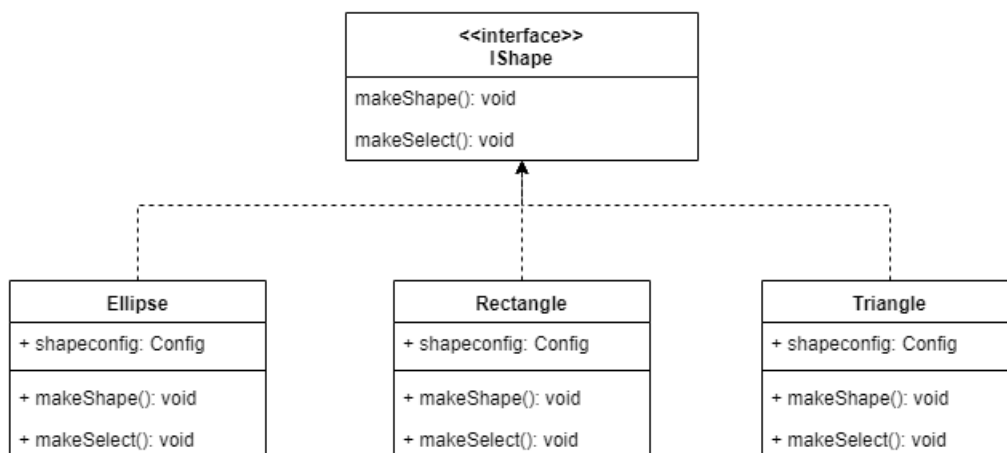
### Null Object Pattern

This program use Config, which is the object that contain Graphics2D, appstate (that automatically include ShapeType, ShadingType, PrimaryColor, SecondaryColor, etc. that in ApplicationState), shapeList, start Point, end Point, and pass this object around in the program. It is the most important object in this program. To ensure that Config object will not have null member in it, the Null Object Pattern is used here. If one of members in Config appear to be null, the constructor of Config will call NullObject class and receive the default member from NullObject class instead.



### Strategy Pattern

When ShapeFactory create IShape object, it will create object of Ellipse, Triangle or Rectangle class which all implement IShape (act like IShape object in ShapeFactory). In these classes, it will have 2 main method, `makeShape()` to draw a shape in chosen shading type and `makeSelect()`, which create outline around shape when being selected.

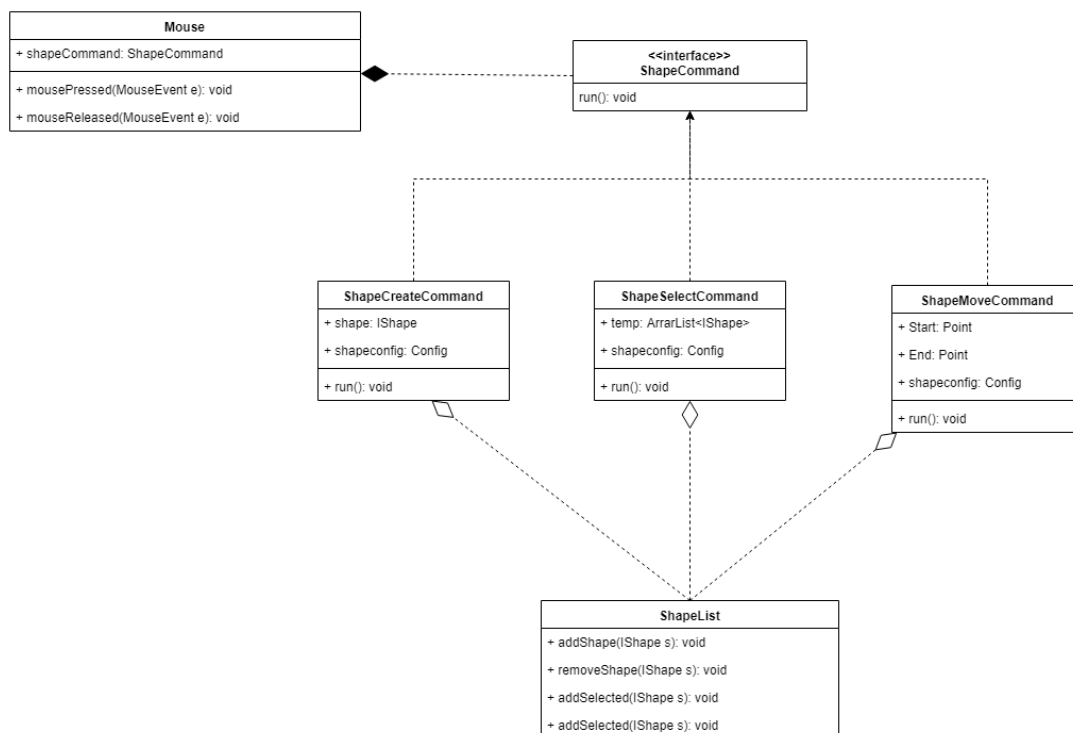


## Command Pattern

This program use 2 Command Pattern

### - ShapeCommand

When mouse release, the program will check what the mode of appState is (DRAW, SELECT or MOVE) and will assign the shapeCommand object to a new ShapeCreateCommand, ShapeSelectCommand or ShapeMoveCommand which all implements shapeCommand and have method run(). Then the program will execute the method run();

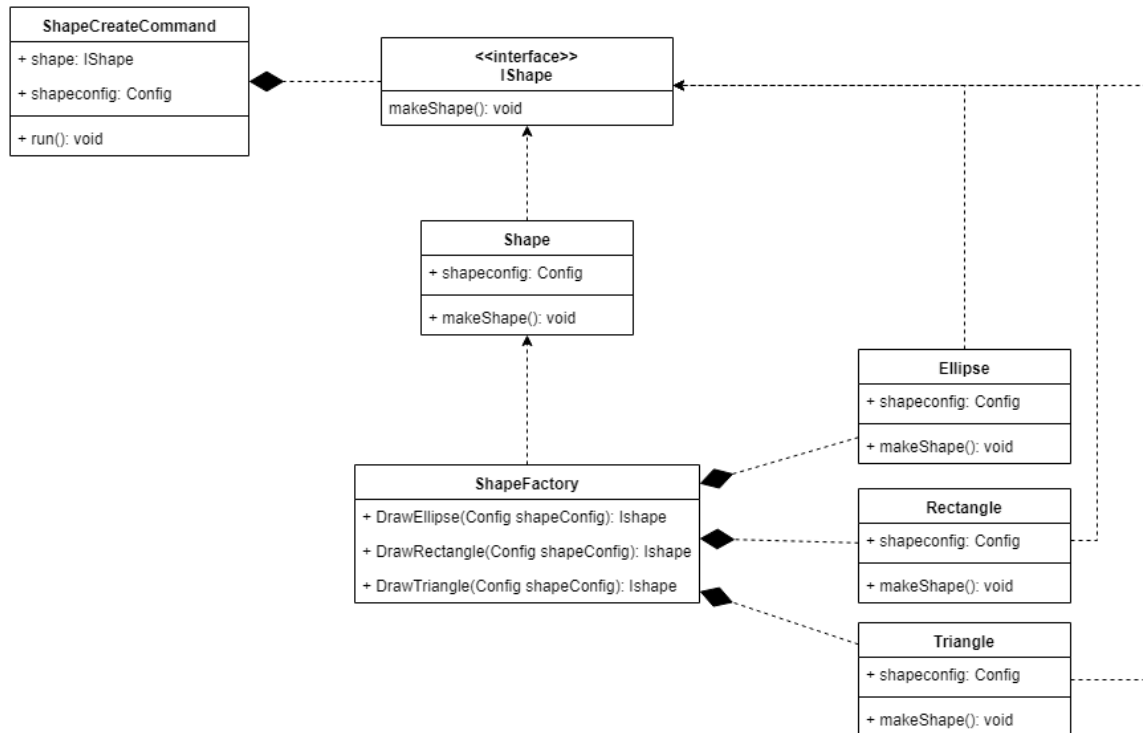


### - IShape

When the ShapeCreateCommand method run() is being called, it will create Shape object that implements IShape and have method makeShape(). The Shape object will call the ShapeFactory method to create the association shape which all implements IShape and have method makeShape();

## Factory Pattern

When Shape is calling ShapeFactory, the ShapeFactory will call the method to create specific shape (Ellipse, Triangle, Rectangle) which act as IShape object.



\* This UML applied to both Command Pattern for IShape and Factory Pattern

## Successes and Failure

This is the first time I really work with Java (I only used to code in C++ in previous Optimized C++ course and C in the past) and I have to study more about its syntax and how it different from C++. At first, I don't even know why we have to use interface or abstract class when we can use concrete object, but this class taught me the importance of them. I can say I'm more familiar with Java and know how to apply some basic design pattern into program.

The first failure of this program occurred in check-in 2, I finish implements the draw part but then realize that it have to be an object of some class to add to the list then it can be 'selected', but it too late and I lost that point. But I can implement it correctly in the end.

The remaining failure is Group and Ungroup function, I try to apply the composite pattern but it complicated and fail to make it work. I also try to simplified and use the List which contain the List of shape (List of List) and work on the outer List as group, but it also fail. I believe that if I have more time and knowledge, I might came back and finish it one day.