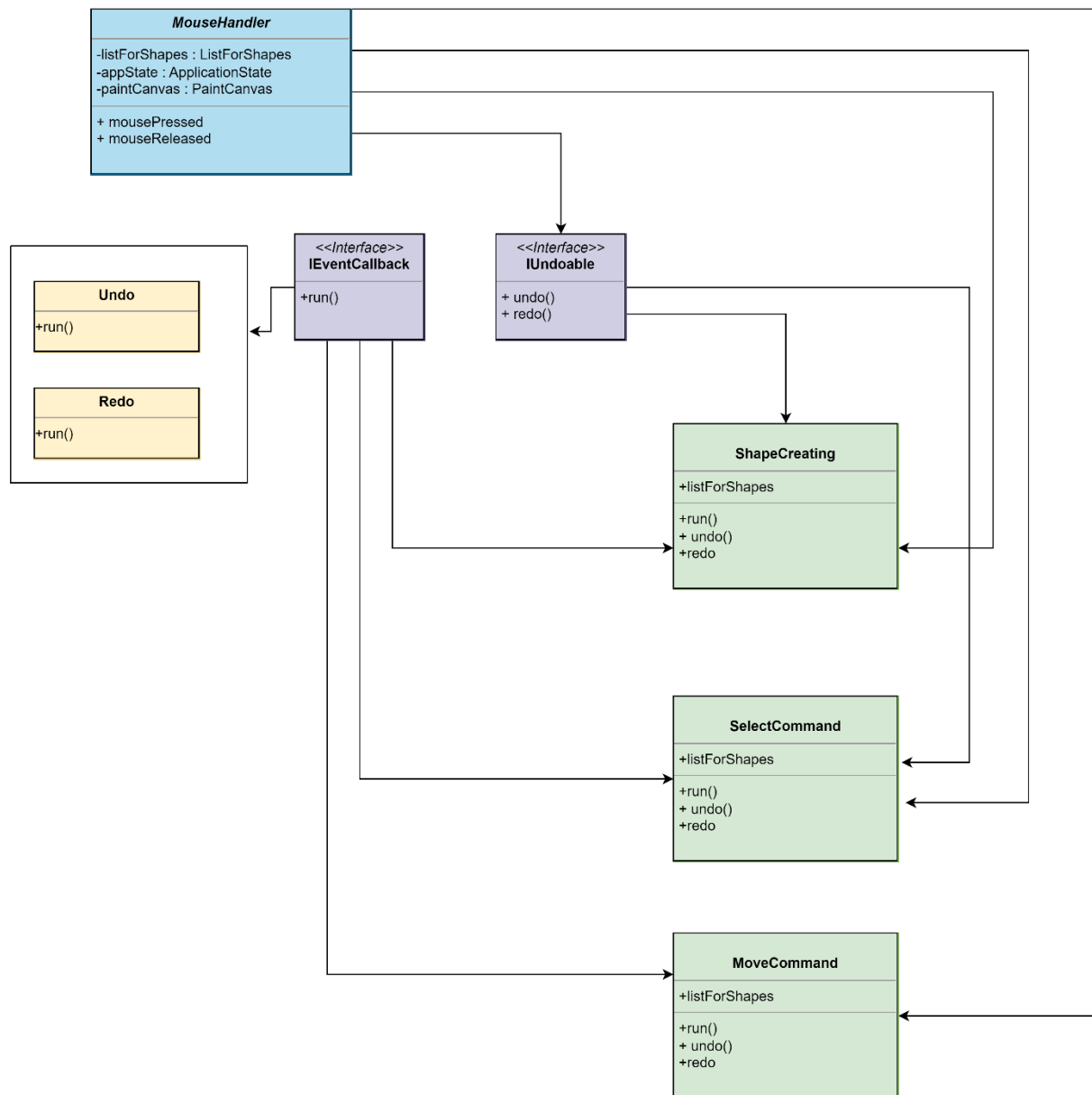


## Command Pattern:

I have used Command pattern to execute most of the command in the JavaPaint application, here I have used the classes like MouseHandler, ShapeCreating, Undo, Redo, MoveCommand, SlectCommand. Because there are numerous commands that must be carried out when a specific action occurs, I decided to use this pattern. since depending on the input or button that is clicked, each command class can implement the IEventCallback interface and use the method in a different way. The problem of what the mouse should perform based on the selected mouse mode was resolved by the command pattern. The draw command is executed if the mode is Draw. Similarly the select command is executed if the mode is Select. The issue of undoing and redoing instructions that had already been executed was likewise resolved by the Command Pattern



## Factory Pattern:

I have implemented the factory pattern to create a solution for drawing different kinds of shapes based on the users selected shape. Here I have used the classes like ShapeDesign, ShapeFrame, RectShape, EllipseShape, and TriangleShape. Factory pattern allows me to generate a variety of shapes that are related to the base ShapeFrame class. Following the factory pattern implementation, the user can also draw ellipses and triangles pointing in the four cardinal directions using mouse click and release placement.

