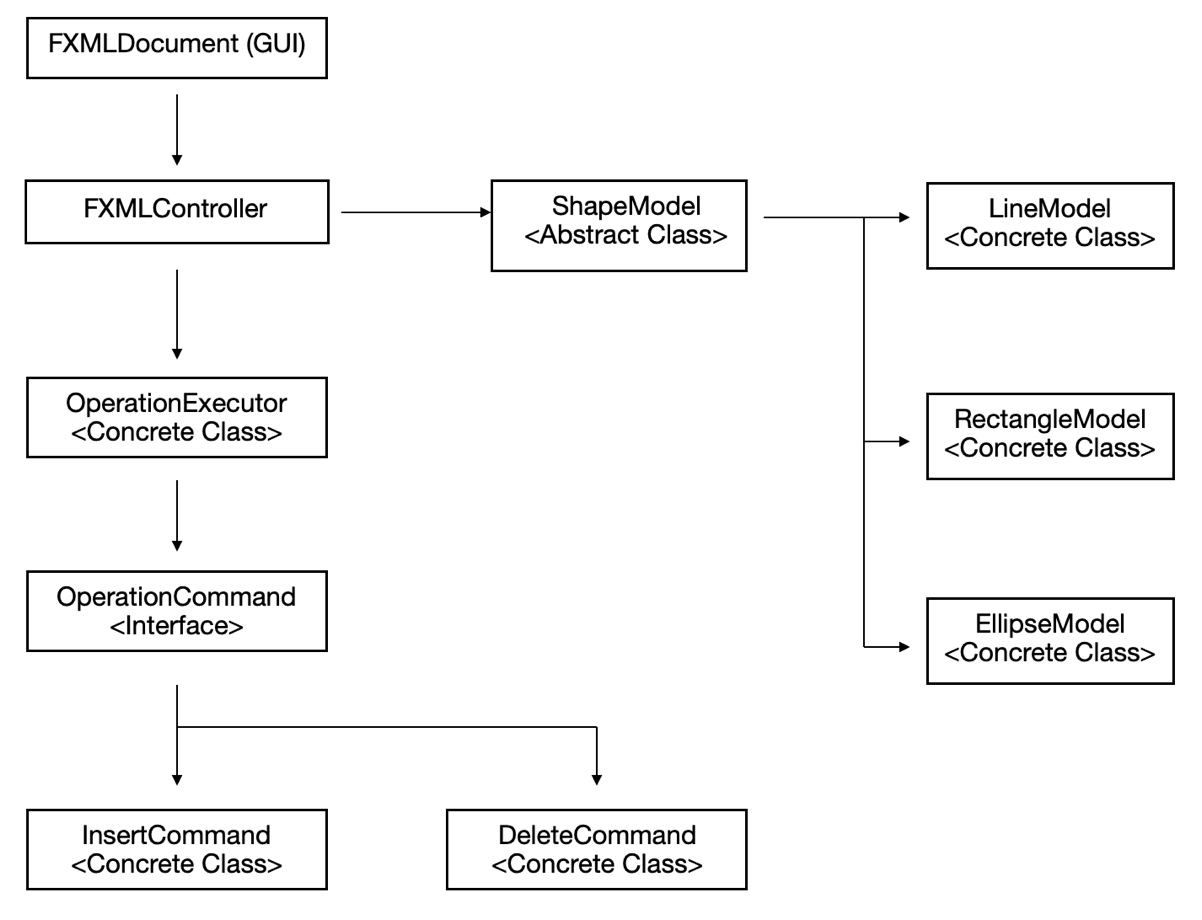
APPLICATION ARCHITECTURE

In the first sprint it was decided to use the Java Swing framework as our project base. This choice was linked to portability problems encountered during the pre-game phase.

Having solved the aforementioned problems, the basic architecture has been modified, mainly replacing Java Swing framework with JavaFX framework.

This change was made to have an explicit ModelViewController architecture for the project.

The view, represented by the user interface, is implemented using Scene builder development tool, it’s represented by the “FXMLDocument” file and contains all graphical elements of the GUI.

In order to achieve all requirements, we designed the raw architecture shown in the following picture (to be further detailed).



All the elements shown in the picture represents a class/interface that we intend to create. This schema will almost certainly be updated during the development.

Specifically, the idea is to integrate two behavioral patterns: “Command” (to keep track of all the operations performed) and “State” (to be able to apply the same methods to different shapes).

For the first, each command will implement the OperationCommand interface and will be invoked by an object of the OperationExecutor class.

For the second, each shape will extend the generic class "ShapeModel", which will contain methods applicable to all shapes.

About conventions:

* The language and the IDE used for the development will be Java and NetBeans
* We will use English language for naming variables, classes, and for comments.
* For classes and methods, we will use camelCase as expressed by the java conventions (e.g. ClassName, methodName).
* Every variable/attribute will follow the methods naming rule (e.g. attributeName).
* Testings will be performed using JUnit (names will follow Junit convention).

Definition of Done

* Task has been implemented, tested as a single module and integrated with already developed task. Previously passed test has been repeated on integrated software.

Estimated velocity: 20 story points