CHOPP

- The Good Boy Interface -

CHOPP
User Manual

Purpose:

This API's purpose is to simplify the process of one's software development as it provides not only a easy mean to create and customize a GUI, but it also gives a frame of a software to work with.

Implementation:

It uses a few elements of swing and awt as support for the graphical and input listeners functionalities, but besides this, everything needed by the API is implemented already.

Most of the classes use a builder pattern to simplify the creation of themselves and to provide a more detailed look over their fields and functionalities.

The Strategy pattern has a purpose too in this API, so understanding of this could be quite helpful in the efficient usage, but it is not completely necessary to use them, as most of the object that use them are only concretizations of more basic, abstract objects.

Usage:

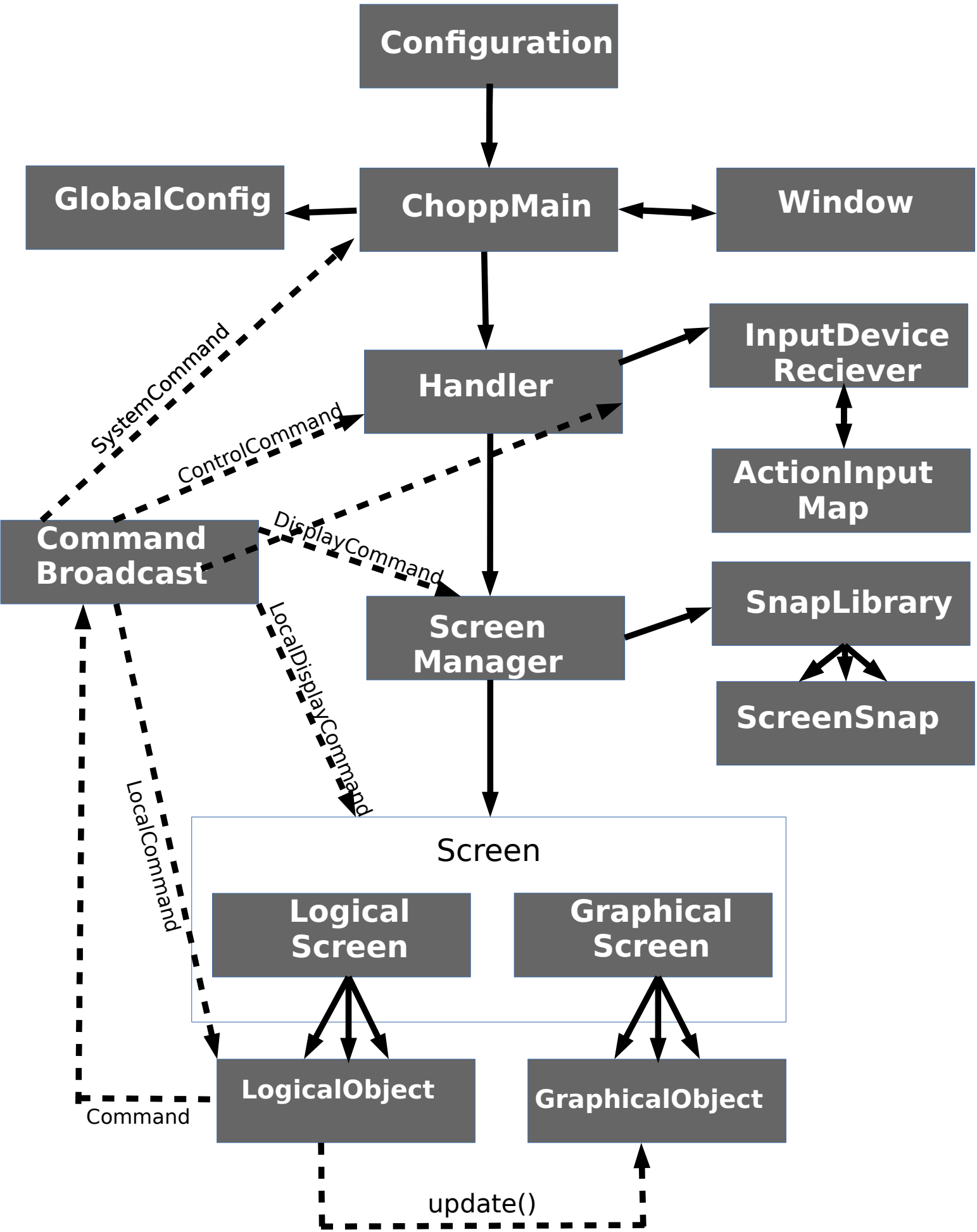
The logical objects serve as elements you act over and the result is mirrored to the graphical object by the call of the **update()** method of the observer patter style implementation and contract declared between those two. Do not think of those objects as different entities, but as different views of the same model, one that encapsulate the logic and functionalities, and the other as the animated face the program render it on the screen.

The API has already a main class names ChoppMain that will take a Configuration instance that encapsulates the details the software should known before it should start running (as for example the dimensions of the window, the screen snaps and the actions as strings that the software will have to work with), and it will open a window by itself according to those configuration details.

Everything in this API works in a hierarchical way, in a chain of responsibility, as the ChoppMain controls the Handler, the Handler

controls the ScreenManager, which controls the active Screens and the pool of Screens, removing or inserting new ones, and every Screen controls the logical and graphical objects it contains.

The commanding aspect of this API is done through a Singleton class called CommandBroadcast which sends the commands back in the command chain, pouring them into the ChoppMain to be executed and send back into the chain.


Configuration
GlobalConfig
ChoppMain
Window
InputDevice Reciever
Handler
SystemCommand
ControlCommand
ActionInput Map
Command Broadcast
DisplayCommand
SnapLibrary
Screen Manager
LocalDisplayCommand
ScreenSnap
LocalCommand
Screen
Logical Screen
Graphical Screen
LogicalObject
GraphicalObject
Command
update()