

COMP303 Assignment6

Actions:

I made Action an abstract class. It has an abstract method `anAction()`. Every Action object needs to override this method to implement its own functionality. Action objects can be created within class `Walle`. A complex action defines its `anAction()` by executing basic actions. The class Action has a boolean field `recharge`. It decides whether check the `battery_charge` or always recharge before execution.

Programs:

Computation:

The client needs to define an algorithm within interface `Computation` to decide if an action is involved in the computation. The program will use `Computation` to check every basic action and return a list of relevant actions.

Logging System:

I used the observer design pattern for this part. A program has a logging system. When an action is executed, it will notify the logging system by calling `statement()`. In the class `LoggingSystem`, the `statement` will print the updated information.