Vending Machine Software Function overview for "Controller()" sales logic

General Process: The Controller software, follows a fairly simple set of behaviours, easily categorized into three groupings, the initialization will run once, but the others may be triggered in any combination.

Initialization behaviour	Coin insertion behaviour	Button press behavior
-Initialize program -Assign client parameters to VM -Register listeners to hardware -Initialize display elements	-Receive coin input -Await coin validity analysis -Increment customer credit	-Determine input validity -Attempt to store coins -Attempt pop sale -Update machine state -Assess state for next sale

Notable Functions:

controller(): Vending machine specifications, requirements, and initial stock(product) attributes are hardcoded into this function, for the instantiation of the machine. It will create and register it's host of listeners, and determine the initial state of the indicator lights, before setting up the timers necessary to run the display.

exactChange(): For each pop in the vending machine the price is fetched and compared with the available change using the checkChange() function. The costs are first checked against mod 25, to determine if low value coins, are available, if needed, for the change. Then checks if the necessary high value coins are available. If either case is true the exact change light will turn on, otherwise it will turn off. The current algorithm is inefficient for standardized prices, but needed to adapt to future price choices.

checkChange(): Is a change returning algorithm, starting with the highest value available coins it decrements the number of coins to reduce the outstanding cost, shifting to smaller value coins when it must. If the machine can return the necessary change the function returns true. Currently the machine will never short change the owner if the change due is less than $0.05, this can be easily changed if required, but our team lacks the legal qualifications on this matter, and resolved in the company's favour.

dispenseChange(): Similar in principle to *checkChange(),* this algorithm will attempt to dispense change, but will not determine if it can prior to doing so. By design it should never run without having earlier run *checkChange()*:

pushButton(): This is the primary function, that initiates most of the systems logic. Though is fairly simple, largely it tests a few key attributes, e.g. coin rack space, input validity, enabled state of machine. Then makes calls to the machine to dispense the desire pop, updates the controllers credit, and then calls to dispense the customers change. In the event of an error, the function will stop any sale and provide proper feedback to the client, or cease function, it determines there is a fault that should not be possible with the given hardware.

System Listeners: As a group the controller's listeners are largely very similar, sharing common functionality in updating the system log of any notable events. Though significant code is also present that is only relevant for test purposes, and can be ignored during implementation.

Misc: The Controller code was also written in crimson text in all editors and IDE's, we hope this shows adequately for you in function.