Eben Schumann

Ben Morledge-Hampton

The Grand Cookie Empire Requirements Specification

The chief requirement of the system will be to manage the methods of cookie production, how they are altered as the game progresses, and how they are presented to the user so that he/she is able to select different upgrades . In essence, this will be done through a menu object which will contain all of the methods of cookie production which will each be objects themselves. The menu will work to help all of these different objects communicate with each other in order to determine net profit over a month. Also, things such as the total capital available to the player, or the total amount of cookies being produced are also maintained by the menu function. Furthermore, the menu will display the prompts for upgrades that will be passed to it by the cookie production objects. The menu will the receive user input as well and pass the results of those inputs back to the cookie manufacturing objects. Then, those manufacturing objects will activate “upgrades” which themselves are objects of a single Upgrade class. When activated, these upgrades alter the characteristics of their respective manufacturing object. Alternatively, production methods themselves may be purchased, which will work similarly. Finally, the menu will also contain a high score table object which will utilize file I/O in order to maintain a list of the fastest times that players have completed the game in.

In the end, the program will boil down to four main class types. First there will be one menu class, which brings all of the other classes together and communicates between them based on user input, as described above. Second, there will be a single “parent” cookie manufacturing class, and through inheritance, the many different forms of cookie production will each be given their own class. Thirdly, each of these cookie manufacturing classes will contain a variety of upgrade objects which all come from one upgrade class whose purpose is to alter the manufacturing object when the menu tells it to do so. Lastly, the class for the high score table will handle all of the file I/O as it restores and saves high scores to a text file. In the event of a new high score it will first compare the scores and slot in the new value where it belongs instead of just being a big mess of scores with no real order.

While the menu will be doing most of the work visible to the user, the general computational leg work will be performed by the production, formerly known as cookie manufacturing, and upgrade classes. As mentioned in the project proposal, each method of production will have four main traits: Rate of cookie production, quality (sell amount) of each cookie produced, cost of production, and risk of some catastrophic failure. Every in-game month, a function in each class uses these factors to determine that month’s production. Mostly, this leads to a single dollar amount being passed back to the menu along with some explanatory text being displayed, the exception being if catastrophic failure occurs. In this event, the other three traits of the class may be altered and additional text will be displayed to the menu to notify the user. Since each method of cookie production has to purchased (except for the first one) these objects will each have their own initial costs and a boolean variable to determine whether they have been purchased and are functioning. This boolean in turn, will determine whether the user is able to purchase the production method’s respective upgrades.

The upgrade objects will contain “modifiers” for the 4 traits of the production object (both multiplicative and additive) as well as a cost variable and a boolean value to indicate whether they are active. They will also contain descriptions of their functions and cost to be displayed to the menu in order to inform the user. When the menu receives user input to activate an upgrade, it modifies its production method and then is no longer displayed to the menu and can not be purchased again. This is achieved by removing it from the vector of upgrades entirely.

The Menu will have a main menu and two sub menus. One sub menu will be the purchasing options for additional production types. They will be purchased by accessing a vector that contains the different production types. When one type is purchased it is then removed from the vector for the duration of the program’s run time. This allows us to present the user with all available production types for purchase without worrying about handling the accidental purchase of an already purchased production type.

The second sub menu will be for upgrades. Upgrades will be plentiful in supply and will be stored in a vector for the same reason that the production types are being held in a vector. If you buy an upgrade it gets removed from the stack and ignored until the end of the program run time. These upgrades are also categorized by the production type they upgrade, and these production types that have been purchased will be contained in their own vector, so that displaying upgrades becomes much cleaner.

There will always be an exit option in any menu to either take you the top level menu or to cycle the next month through and bring up a new month’s top menu with new values for player money and maybe a new value for cookie production per month, CPPM.