Nicholas Skinner

Assignment 6

CS260

HASH TABLE, SMART AND DUMB

Design:

Initially, I had no clue where I should start for this project, I knew the end result that I wanted, but I didn't know what approach I should have to it. After looking through some tutorial videos to get a better idea of what I was doing. The most challenging part of the design for me was, how was I going to assign the strings of letters to a position, I knew how I could do it with integers, but I improvised a method of creating a number with respect to its location from a in the alphabet, I appended these into a string and then pulled a number from the string (i.e. aaa becomes 111, bbb becomes 222, etc).

My other big hurdle while creating this was the logic behind my collision detection/ resolution. I wasn't satisfied with just replacing value for my collisions, so after finding some information on stack overflow I had built a method to search in the next value that is blank in the array and populate there. The only problem with this method is when you want to submit more characters than the array is long. An error will occur, and I have no resolution for this, as I don't have a backup collision detection to replace after it can't find another position to put an item.

Testing:

Known bugs:

Too many letters	If there are more than 3 letters in an object in the array, my code will throw a fit. I have a hardcoded limit of 3 or less items, and more will cause errors.
Non-letter item	As I have my code find alphabetical position, numbers and other ascii characters will not play nicely with my code, I believe I had tested with a few numbers and had ended up getting a negative where it doesn't belong (something like a8b became 1-472) said example could not be formed into an integer as it is not a number.
Too many items in initial array	My array has a somewhat hardcoded limit of 30 items, if more appear it creates conflict and cannot resolve it, as I do not allow user input this is not as big of a deal, but it is still of note.
Missing value in one (if not many) array slots	Populated a -1 in the place that it would have some other value, I think I have this hardcoded in, but I should probably edit this.

Reflection:

I'm noticing more and more as I build these bigger projects that I'm leaving comments for individual items as basic as a print to try and get all of my logic out there. While having a friend who had taken the class previously look at my code, he said that it was much easier to look through because I am explaining every loop's function, every out of method call, and every methods reason to exist. I feel that this is very important in moving toward a more team-oriented real-world platform where others may have to look at your decade old spaghetti code and figure out what it did, comments can help speed up this process (unless someone previously updated the function, but not your comment).