Eduardo Nava

CS202

Project 4 Documentation

The purpose of project 4 was to recreate our previous projects using pointers, and dynamic memory. The requirements were the same as for project 3 except we were required to do the mini challenge. Which was to output the configuration as it would appear in real life and include the other data stored within the symbol struct. The overall purpose was to create a program that would mimic a slot machine's functionality.

The design of the program was similar to the design of project 3. In this case I decided to reuse many of the functions that I used previously. The first function that was to be modified was the function that would read in the symbols file to a symbols struct within the program. This was the only function that required memory to be dynamically allocated within the function itself. The variables from the function that required dynamic memory were a filename array, and a character input array. The filename variable was needed to hold the file name specified by the user, which would subsequently be used to open that file within the program directory. The other variable was a char array that was required to hold the string being read in from file and acquire its string length. This string length would then be used to allocate the exact amount of memory required for the symbol filename member. From there the loop took care of everything and the memory for the filename and char array variables were deallocated immediately before the function's completion. The next function to be written required generating a random configuration for the program. This was accomplished using two loops and a random number generator that would be called within the innermost loop to ensure every symbol within the reel arrays was randomized to some extent. After both of the loops completed, the affected pointers were reset back to their base address to ensure that they would be reusable for the next time that they would be called in a function. Next the print configuration function was needed. This function required modification in order to incorporated the format required by the project 3 mini challenge. This was accomplished by using 2 loops which would print the data reel by reel for each stop. The final function was a function that would allow the user to display the data for any stop and reel number chosen. This simply required 2 int data types to hold the chosen location then a few loops to reach that location. From there the specified data would be printed and the function would end.

These functions operate within a menu system that runs until the user exits. Once the user exits the menu, then the program deallocates the memory for the reel and symbol arrays before exiting. The one problem I had was getting the random config generator to generate a random number for every reel. The rest of the functionality works completely, but if I had to change anything, I would want to increase the number of symbols within the symbols file to have a wider variety of symbols for the program to choose from.