NOT FOR A GRADE
bottles

Bruce Cain

June 2018

1 Design Document

Using a function to sing 99 bottles of beer on the wall too help control input to the number of bottles. Also to set up which beverage to count through out the song.

2 Targeted Features

Implementation of random number of bottles between 1 and 99 and giving the user the option to do so. Giving the option to accept two arguments for the number of bottles and the beverage type. Including the processing time that the program takes. Using the numToWord() function to output the word of a number instead of the number itself.

3 Architecture

Using a function for the song to keep the main function clutter free and keep from repeating any code. In the bottlePrint() function, using a dictionary to handle the printing of the song. The idea was to keep from having to create multiple variables or prints with the same line with a single letter difference. In the function numToWord() a dictionary is used to hold all the numbers required to play the song within the constraints given.

4 Testing Procedure

Testing the program was done through continuous and manual execution via the command line.

5 Write Up

5.1 Goals

Create a program to output the classic 99 bottles on the wall song.

5.2 Challenges/Success

Challenge that came up on the architecture was how is it possible to not repeat the song several times but make it work with plural and singular bottle. Accepting multiple command line arguments to control how many bottles and what beverage the user wants. Added the ability to handle Ctrl+C and Ctrl+D at the appropriate time in the program. Finding a way to handle getting two numbers and getting the correct placement of the numbers. An example would be make 21 to twenty-one.

5.3 Lessons Learned

Using a dictionary may not be the optimal way to handle print statements because of readability. I have tried to make it more readable with the dictionary but in the future try to implement a print statement that's more efficient and readable. After modifying bottlePrint() to not use the dictionary and managed to use only one print statement with two if checks. Thanks to Ben Miramontes for pointing out that repeating use of dictionaries are best used as a global variable. Which was put into practice for the toWord() function which helped with performance.

5.4 Conclusion

The project was a good entry level problem to the course. Used the majority of what was covered in the first week. Looking forward to future projects and improving.