The Magical Java 8-Ball

Submitted by:

VILLENA, Andrea Dominique O.

BS Information Technology – NW1H

Submitted to:

THOMAS ANTHONY D. SANCHEZ

Instructor, CTCC0213

I. Background and Description

The Magical Java 8-Ball is a desktop application based on the original Magic 8 Ball and can answer any yes/no questions that the user inputs in Java. It allows the user to experience fortune-telling about their future or decide for themselves.

The twist is that this Magical Java 8 Ball will give brand new answers that gives a more accurate and balanced result compared to the original. It can answer up to 14 new positive and negative answers or decide otherwise with 7 non-committal answers that forces the user to ask again. The original Magic 8 Ball has a bias with 2:1 affirmative answer over the negative answers while the Magical Java 8 Ball gives an equal 1:1 and an included 1:4 chance of an evasive answer.

The desktop application is capable of conducting the following functions:

- Answering any questions that the user inputs with a positive or negative answer ranging from serious to sarcastic.
- Decides to randomly give non-committal answers.
- Knows if the user sent nothing in the question prompt.
- Will ask if they like to try again that loops until they decide to terminate the program.

However, the desktop application is not capable of conducting the following functions:

- To give the same answer again to a specific question.
- It is not reliable to give advice on serious and important decisions.
- It is not psychic or has the supernatural ability to predict the future as it operates by RNG.

The application is based on the Magic 8 Ball that is created by Mattel and should only be used for entertainment purposes.

II. Program Logic Flow

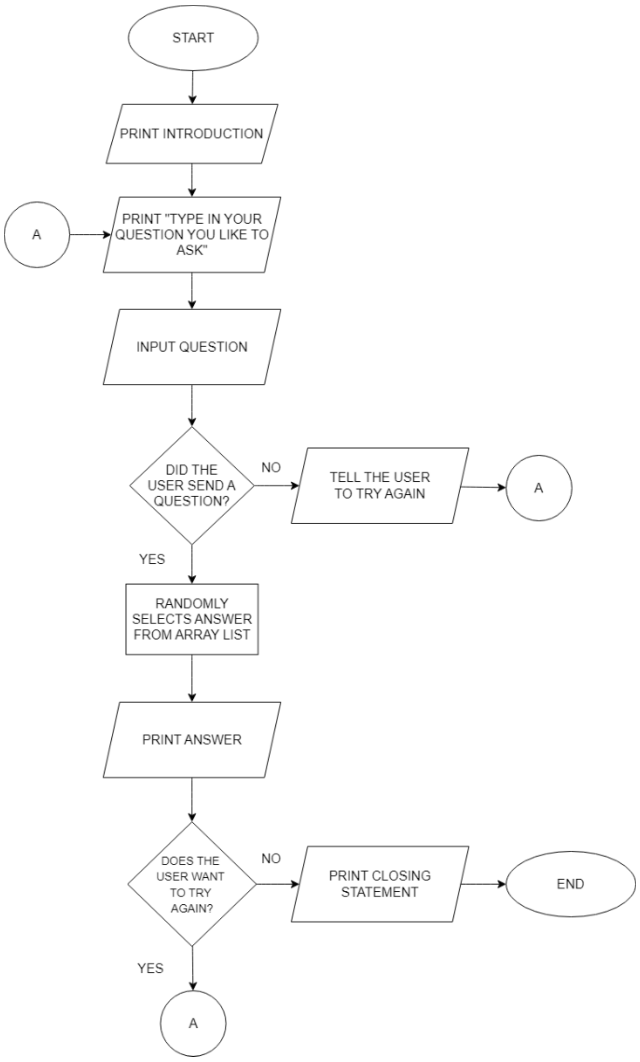
START
PRINT INTRODUCTION
A
PRINT "TYPE IN YOUR QUESTION YOU LIKE TO ASK"
INPUT QUESTION
DID THE USER SEND A QUESTION?
NO
TELL THE USER TO TRY AGAIN
A
YES
RANDOMLY SELECTS ANSWER FROM ARRAY LIST
PRINT ANSWER
DOES THE USER WANT TO TRY AGAIN?
NO
PRINT CLOSING STATEMENT
END
YES
A

Figure 1. *Program Flow Chart*

Figure 1 illustrates the logical flow of the Java console program. The program begins with the declaration of the scanner, random, loop, and allanswers. Next, it starts the process block of printing the introduction, and another block that asks for their question that is connected to the A connector for the loop. In which the input process block prompts the user to input a question.

Then, through a decision block, the data will be evaluated if the user inputs a question. If they leave it empty, it will promptly be asked to try again and be looped again to the A connector. Otherwise, the process block will start the randomization in the list of allanswers and the result will be printed with the output block.

After the results, it will prompt you to answer again with a decision block. If they answer yes, it will continue the loop and return to the A connector. But if they decide to answer no, they will be greeted with a final closing statement and terminate the program.

III. Repository

Instructions: Create a GitHub repository for your desktop application. Ensure that all files (i.e., Java, JPEG/ PNG) are included in the repository. Name your repository in this format: *desktopappname-section-ctcc0213*. For instance, you have created a food delivery application, repository name shall be like this: *fooddelivery-sd1a-ctcc0323*. Make sure it is a *public* repository. Include the username and repository name on the space below. Replace the blue text.

Repository Name: **AndreaDominiqueVillena/TheMagicalJava8Ball-NW1H-CTCC0213**