

CISC 110 Lab 7: Magic 8-Ball

Finish implementing the Magic 8-Ball script. Magic 8-ball is a sphere that's used for fortune-telling or seeking advice. It will tell you whether something that you ask is true now or will be in the future by giving you random answers that are affirmative, negative, or non-committal.

Download Magic8BallStart.fla from our web page. The starting script in the file Magic8BallStart.fla lets you "shake" the 8-ball by dragging it back and forth on the screen. It always results in the same answer. You will complete the script in order to make the 8-ball give a variety of random answers.

To complete the script, you need to fill in or change ActionScript everywhere within the script that you see "YOUR CODE GOES HERE" or "CHANGE THIS". First you will make the 8-ball display a variety of random answers, instead of always the same answer, by randomly selecting answers from an array of possible answers. Then you will make it check whether it has already used an answer, and keep selecting random answers until it chooses a new one, so it doesn't keep repeating itself.

Here's a summary of what you need to do:

1. Write a single line to call the randomAnswer function, which selects a random element from the answerList array, a list of all the possible answers. Make the call within the giveAnswer function, which is the handler for when the user stops dragging the ball back and forth. Put the result in the variable called answer. The giveAnswer function already displays answer in the Magic 8-ball answer text field. Now the 8-ball should display a variety of answers.
2. In the giveAnswer function, write a single line that uses the Array class push method to append the answer that's been chosen to the end of the usedList array. See pp. 170-171 in the ActionScript text for a similar example.
3. Write the usedAnswer function, which checks whether an answer has already been used by checking whether it's in the usedList array. It returns true if the answer is in that array and false otherwise using a for loop
4. In the giveAnswer function, write a while loop that keeps calling the randomAnswer function to return another random answer. It keeps looping as long as the answer has already been used, which it checks with the usedAnswer function. Once you have the basic while loop working, modify it to only keep looping if the number of answers in the used list is less than in the answer list, to avoid an infinite loop.