Department of Computer Science & CINSAM NKU Summer Programming Workshop 2015



Project 7: Who Wants to be a Millionaire?

Implement a variation of Who Wants to be a Millionaire. Start with the skeleton code on the website. The basic game works like this:

- 1. Create two arrays, a String array called *questions* and a char array called *answers* where the two arrays will be the same size. The answer for questions[i] is stored at answers[i], that is, the same array index is used to get the question and the answer. Use multiple choice questions only for your *questions* array, with 4 possible choices labeled A, B, C, and D. Have at least 15 questions to start. Expand the questions/answers later after you get the program running.
- 2. Use a while loop to control the game as follows:
 - a. loop until the user misses a question or enters 'Q' to quit.
 - b. Randomly pick a question, i, and display the question to the user.
 - c. Get the user's answer as a char ('A', 'B', 'C', 'D', or 'Q' for quit).
 - d. If the user's answer matches answers[i] then the amount the user has won is doubled, otherwise the game ends (exit the while loop) and the user loses all of their money. Match the user's answer as if(userAnswer==answers[i]).
- 3. Outside of the while loop, output the amount the user has won.

Enhancements:

- 1. The first enhancement is easy, keep track of the number of properly answered questions and output this along with the amount after each successful question and once the game ends.
- 2. The second enhancement is also easy. The game ends automatically if the user has answered the 15th and final question. So exit the loop if the user misses a question, answers question 15, or selects 'Q'.
- 3. Add a Boolean array called *used* which is the same size as your questions and answers arrays. Initialize all of its elements to false. Once you have used questions[i]/answers[i], set used[i] to true. Now, when you randomly pick a new question, i, make sure that used[i] is false. Use a do-while loop for this. For instance: do { i=g.nextInt(...); } while(used[i]==true);
- 4. Add an integer variable called lifeline and set it to 3. The user is allowed to use up to 3 lifelines. The user may request a lifeline (use 'L' for the input character for a lifeline) in response to the current question as long as he/she hasn't used all 3 lifelines yet (as long as lifelines > 0). For all 3 lifelines, we will use 50/50s which removes two of the wrong answers. This will require that you display the answers with 2 removed. To implement the 50/50 option, create another array which, for each question, lists only 2 choices (including the correct choice). For instance, if questions[i] is "Who was the first president of the United States" and answers[i] is "A. George Washington B. Thomas Jefferson C. Abraham Lincoln D. Barrack Obama" then add an array called fiftyfifty where fiftyfifty[i] will be be "A. George Washington C. Abraham Lincoln" (having removed B and D). So, after the user has entered 'L', you would test to make sure lifeline > 0 and if so, display fiftyfifty[i] and decrement lifeline. If lifeline == 0, output an error and ask the question to re-enter their choice of A-D.
- 5. Organize your questions into three questions arrays. The first set of questions is a group of easy questions, the second set is a group of medium questions and the third set is a group of hard questions. Use the easy questions for the first 5 questions asked to the user, the medium for the next 5 and the hard for the last 5 (or use 6 easy, 6 medium and 3 hard).
- 6. Change the lifelines so the user can request phone a friend (E), 50/50 (F), or poll the audience (G). Create arrays for the response to each type. For instance, for poll the audience for the president question above, you might have audience[i]="A. 82% B. 10% C. 8% D. 0%". If you do this, then instead of having 3 lifelines, they get one of each of these. Make sure that some of the responses in phone a friend and poll the audience are wrong or somewhat misleading just so that lifeline doesn't always lead to the right answer.