Homework 3

"Vue.js and Google Cloud Firestore"

The following problem set is worth 100 points total. Please submit a zip file containing all relevant code to the Canvas drop box by the due date of the assignment.

Requirements

For this problem set, you are building a game using *Vue.js*. You will store your high scores in a *Google Cloud Firestore* database. As to the game that you will be building:

Who wants to be a Millionaire?

Game Details

Who wants to be a Millionaire is a television game show where players are given a series of multiple-choice questions to win cash prizes. The game has been on the air for over twenty years and has had its format modified a few times. For instance with the original British version of the game, the player was given 15 questions. With each correct answer, the player could win the pound (Sterling) amount associated with the question:

Question number	Question value
1	£100
2	£200

3	£300
4	£500
5	**£1,000**
6	£2,000
7	£4,000
8	£8,000
9	£16,000
10	**£32,000**
11	£64,000
12	£125,000
13	£250,000
14	£500,000
15	***£1,000,000***

If the player incorrectly answered any of the questions, the game was immediately over. You'll note in the table above that there are three shaded levels where the player was guaranteed some earnings. For instance, if the player got the sixth question incorrect, the player at least can walk away with a thousand pounds.

North American Version of WWTBAM

The North American version of the game brought a few changes that the British version does not have (initially). For instance, time limits were added to each question. In the British version, the player could take as long as needed to answer the question. With the North American version, a timer was added as follows:

Question number	Time limit
1–5	15 seconds
6–10	30 seconds
11–14	45 seconds
15	45 seconds (+ any accumulated remaining time from the previous 14 questions)

Game Requirements

For this assignment you are to implement a playable version of WWTBAM with *Vue.js*. In the television show, questions are presented in the following format:
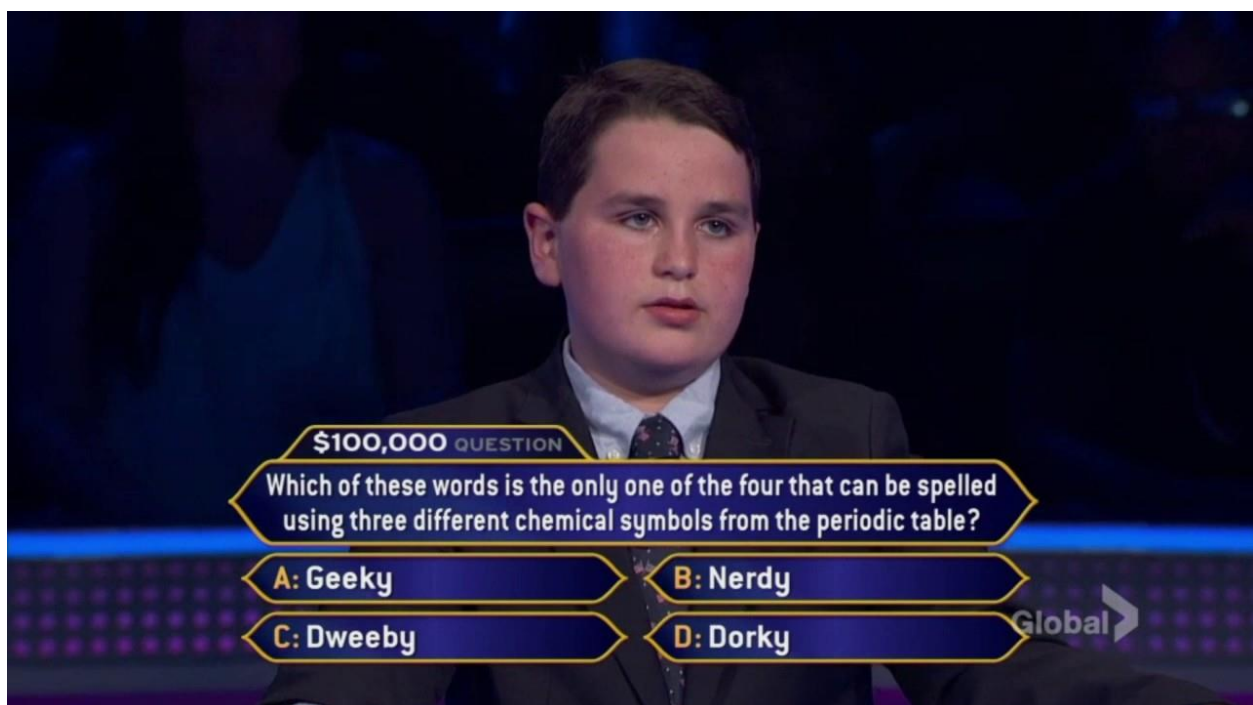
You'll note that the player can either select A, B, C or D when answering questions. For your assignment, you'll need to handle **keyboard events** in Vue. That is if the user selects A, answer A is chosen et al.

Implementation

A strategy for you to build this game is as follows:

1. Store a list of questions and answers in a static data set in your code.
   a. If you are able to find an API with data, feel free to use it
2. When the player starts the game, collect the player's name.
3. Start presenting questions one after another to the player.
4. Allot a timer to each question. (The timing strategy is up to you how you want this to be implemented).
5. Allot minimum prize earning levels (for instance the user is guaranteed money after answering 5, 10 and 15 questions correctly in the North American version).
6. The game ends when either:
   a. The time expires on a given question.
   b. The player guesses the question incorrectly.
   c. The player guesses all questions correctly.
7. Once the game is over, record the player's name and their prize earnings in a *Google Cloud Firestore* database (that you can create for free).
8. Show the user a list of players and their "high scores" once the game is over.

You do not need to implement any of the additional features that helped popularize the game; for instance, "Phone a Friend".

Submission Requirements

For the submission, submit a zip file with all assets needed for me to run your game. If you are using the Vue.js CLI for creating a standalone application, please do not include your *node_modules* directory in your submission.

Submit using the Canvas drop box before the due date.