

First Report

“Quote Unquote” Trivia App

Sirona Miulescu

Computer Science in English, Year 2, Sg. 1

Programming for mobile devices

Submission date: 19.05.2021

Supervisor: Liviu Octavian Mafteiu-Scai

1. Abstract

The aim of this report is to give an overall view over the proposed app, bringing information about the target users, state of the art and some information regarding the plan of development.

2. Goal and users

The goal of this project would be to develop a functional application using Android Studio. The app would fall into the entertainment category, being a trivia app which would test the knowledge off well known quotes. The target user would be the people who have some of a knowledge in the domain, meaning they are interested in the important social gatherings , like to watch movies, aware of the important events and personalities from throughout history(Ex: Einstein, Nelson Mandela, Gandhi, Marilyn Monroe). The questions would not be related, a vast range of events being taken into account.

3. Contribution

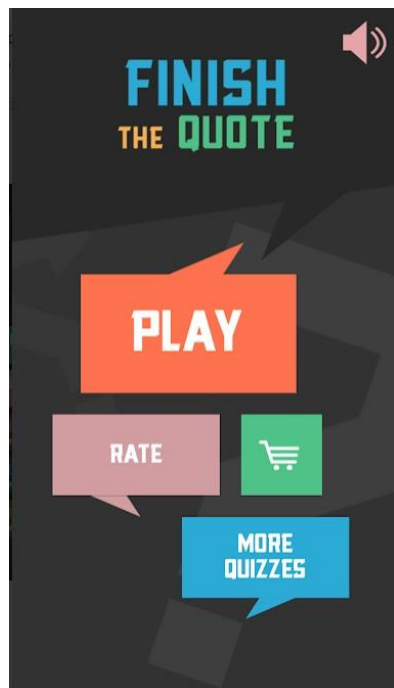
Due to the fact that this is not an original idea, since similar applications exist, the subjects approached and the design would be the differentiating points and my contribution. Also, the combinations in which the code and the design exist is also the contribution of the developer.

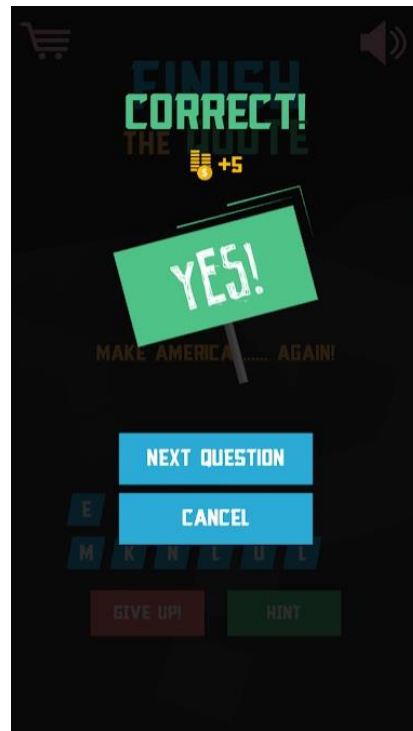
4. Introduction

As an introduction to the app, as all trivia applications, “Quote-Unquote” will test the knowledge of some domain through some questions which have four possible answers but only one correct one. Upon opening the app, a start button will be visible and once you click on it, the questions will start appearing, one after the other. After selecting the answer the user thinks is correct, by simply clicking on it, the app will provide check the correct score and move to the next question

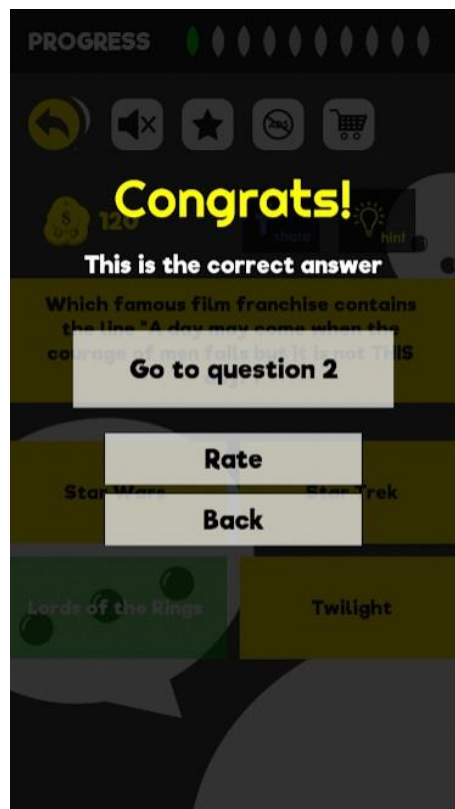
Some of the other trivia applications which have the same idea, of social testing.

“Finish the quote”

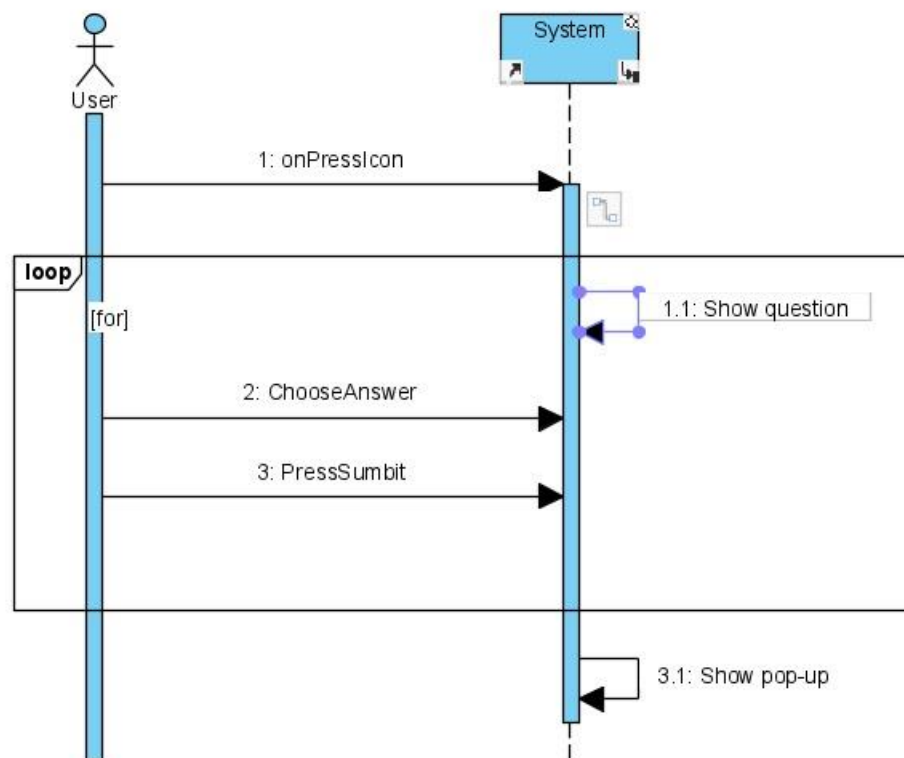




“Who Said That”- Movie Quotes Quiz Game



5. Functionality



6. Running the app

Since this was my very first application, the idea is simple and very straight forward to use. Upon clicking, the questions start appearing in order. For the user to continue, he must select an answer and click on submit. No submission is accepted if a button is not clicked. After the final question, a pop-up appears, presenting the score and the possibility to play again.

7. App's structure

The base of the project is the class `Question` with attributes named suggestively. The only method in this class is a Boolean returning method which checks whether the players answer is the correct one.

Most functionality is developed in `MainActivity`. I will explain each function individually:

- The attributes of the class, named accordingly, are declared to keep a state of the game throughout the play.
- After the starting of the game, we execute **onCreate**. Here, we show the icon of the game in top, left corner, before the name of the game, we set some **OnClickListeners** to all of the buttons and instantiate the views, to help us print the questions later.
- The method **showQuestion** does exactly what its name suggests. It sets the text of the views with the values received from a **Question** object received as parameter.
- **displayQuestionRemaining** only sets the value of the attribute equal to an integer received as parameter
- **onAnswerSelected** we create a new **Question** object and set the value of the **playerAnswer** attribute equal to that received from when we clicked on the button. Based on this integer, we choose how to display the answers. For each option, we set all the text values accordingly.
- **OnAnswerSubmission**: we first create a new **Question** object which is of help in calling other methods. We first check if the player actually chose answer before submitting it. We then check if the answer is correct, and if it is, we increment the attribute used to retain the correct nr of questions answered. We remove the current question from the array, to not print it twice. We then check the number of questions left, and if there are any, we choose the next question, otherwise we present a pop up.
- **chooseNextQuestion** is the method that picks an arbitrary question to be displayed next.
- In the method **startNewGame**, we declare and initiate an array of questions, we choose and print the first question, and reset the number of questions in total and answered right.

8. Conclusion and future work

Since it has been my first AndroidStudio project I have many things to improve in it so a multiple activities would be nice. The sleekness of the design could also be improved.

9. References

1. Udemy
2. [Android Developers](#)