

GeoQuiz Model-View-Controller-05

In this assignment you will enhance your GeoQuiz app to allow for more than one question and provide the user with a next button to navigate through the questions. There is an image on the LMS and at the bottom of this lab that shows what the final app should look like.

1. Open your “GeoQuiz” app, go to GitHub and create a new repository for this lab, then connect your GeoQuiz app to the new repository and perform a push.
2. Add a new class called Question to your project
3. Add two class variables to your Question class, one will hold the question string resource id (int) and the other will hold the answer (true or false, therefore boolean)
4. Add a constructor to your Question class that will take an int and boolean parameter and set the values of your class variables
5. Generate setters and getters using Android Studio’s automated setter/getter generator
6. Add a new button to your layout which will say “NEXT” and will be used to move to the next question
7. Edit your string resource file and add 5 different question strings
8. Add a class variable in your QuizActivity class that will hold an array of questions and initialize it to hold the 5 questions your created (e.g. create 5 new instances of your Question class and set the string resource IDs and answers for each)
9. Add a class variable in your QuizActivity class that will hold the current question’s array index (default the value of this variable to 0)
10. Add a class variable in your QuizActivity class that will hold your TextView and in your onCreate method, hook this variable up to the actual TextView (e.g. using findViewById)
11. Add a class variable in your QuizActivity class that will hold your next button and hook it up to your button
12. Write a method called updateQuestion in your QuizActivity class that will update the TextView to show the question text of the current question (hint: you can do this by using the question index variable to pull the current question’s string resource ID from the item in the array, then use that string resource ID to setText on your TextView)
13. Call your updateQuestion method from your onCreate method to ensure the first question is shown when your QuizActivity screen is displayed
14. Now create a click listener in your QuizActivity class on your next button. In the next button’s on click method, write code to increment the question index to the next one, and call your updateQuestion method
15. Write a method called checkAnswer that will accept a boolean (this will be the answer the user pressed, e.g. true or false). In this method write code that will check the answer with the current questions answer and show a Toast message if they were correct or not
16. Update your true and false button click listener’s on click methods to use the checkAnswer function instead of the code from the previous lab
17. To make your app look better, put an image of an arrow on your next button. First download the images from the LMS, then add them to your resources. Next set the drawableRight attribute of your button in your layout file
18. Run your app using the emulator and ensure it works properly

