GitHub Username: Boris Starovoyt (https://github.com/bolf)

Trivia Quiz

Description

Trivia Quiz is a free quiz-app. It offers different questions of many categories. It can become a good tool for broadening one's horizons in various spheres.
For user's convenience flexible settings and analytics are implemented: user can set quantity, difficulty and categories of questions he wants to answer and get the data of his own achievements.

The app utilizes Open Trivia DataBase https://opentdb.com. This service kindly provides free to use, user-contributed trivia question database.

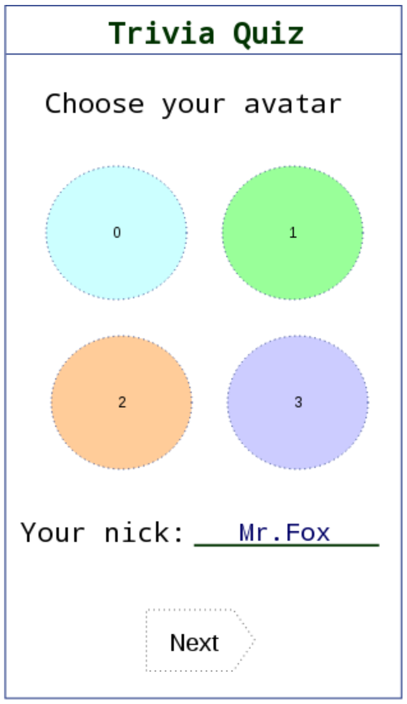
Intended User

An intended user is anyone who likes quizzes and has a bit of spare time =)

Features

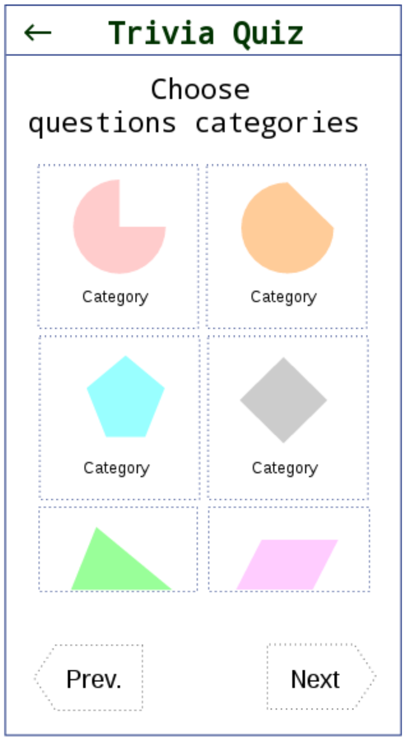
- Trivia Quiz offers flexible settings (mentioned in the Description paragraph) and intuitive UI
- Tracks and demonstrates user's achievements
- Utilizes Android Architecture Components
- Follows the principles of Material Design

User Interface Mocks



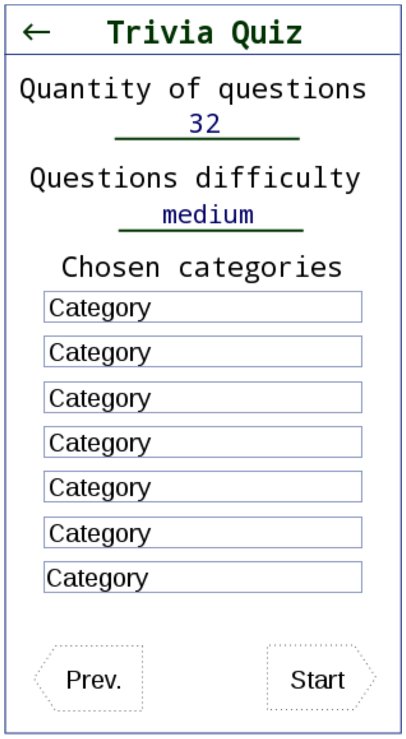
Screen 1

Screen 1 (UserSetupActivity) is opened for the app's first launch. Also it is used for changing avatar and user's nick. (Instead of just colored circles there will be pictures inserted.)
This activity forms and edits the user instance's basic properties (sets nick and picId which in conjunction represent user's unique identificator).



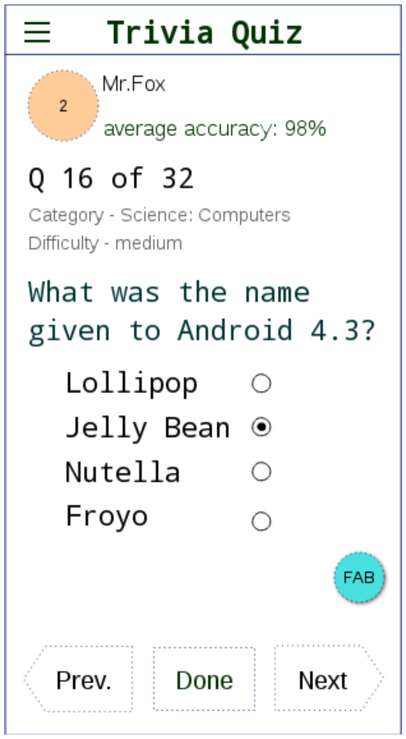
Screen 2

Screen 2 (ChoosingCategoriesActivity) serves for choosing the set of categories for quiz. Chosen set of categories will be used as filter for getting questions from the server.
To choose interested category user's got to tap category's representation in the grid (category representation's background will be changed in response showing that category is chosen).



Screen 3

With screen 3 (QuizSetupActivity) user is offered to set the quantity of questions and questions difficulty. Also list of previously chosen question categories is shown (only titles without pictures).
It's a kind of confirmation activity before start also.



Trivia Quiz

Mr.Fox

average accuracy: 98%

Q 16 of 32

Category - Science: Computers

Difficulty - medium

What was the name given to Android 4.3?

Lollipop

[✓ Jelly Bean]

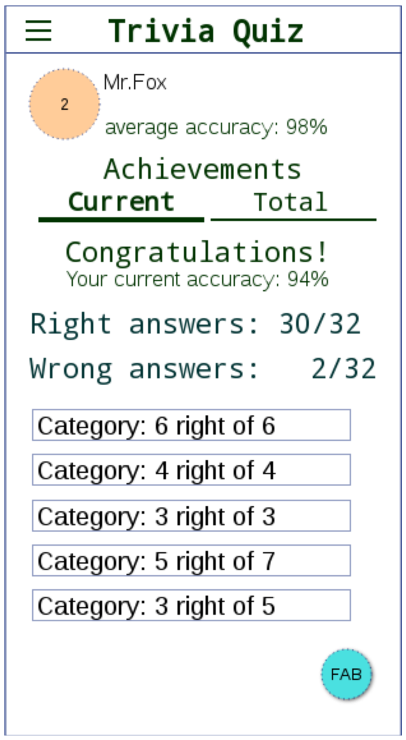
Nutella

Froyo

Screen 4

Screen 4 (QuestionActivity) is for working with questions. App starts it after user hits "Start" button on the previous activity. "Done" button stops the quiz and brings up the screen 5 activity (showing the user's achievements).

Also floating action button is provided which lets user to share current question with his friends =)



Trivia Quiz

Mr.Fox

average accuracy: 98%

Achievements

Current Total

Congratulations!

Your current accuracy: 94%

Right answers: 30/32

Wrong answers: 2/32

Category: 6 right of 6

Category: 4 right of 4

Category: 3 right of 3

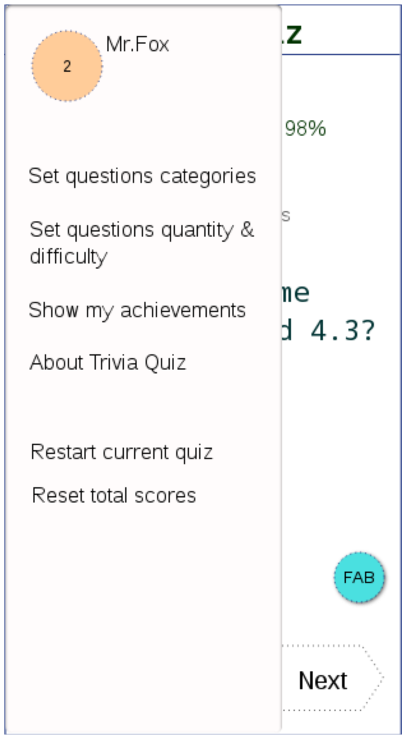
Category: 5 right of 7

Category: 3 right of 5

Screen 5

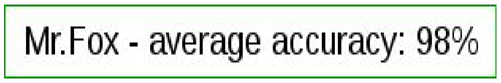
Screen 5 (ScoreActivity) the achievements activity, representing user's scores. The main UI control here is ViewPager with 2 pages, showing the last passed quiz's results and the accumulated results for all previous passed quizzes.

Also floating action button is provided which lets user to share his scores with his friends =)



Screen 6

Screen 6 - navigation drawer panel that shows app's main navigation menu. It is hidden when not in use, but appears when the user swipes a finger from the left edge of the screen or, when at the top level of the app, the user touches the drawer icon in the app bar. The panel contains links with self-describing captions.



Screen 7

Screen 7 - home screen widget. Shows current user's answers accuracy. Widget's data is updated automatically when user's average accuracy changed. Tapping the widget opens Trivia Quiz App.

Key Considerations

How will your app handle data persistence?

Trivia Quiz will use Room persistence library in conjunction with other Android Architecture Components.
Also shared preferences will be used for holding the current user's data.

Describe any edge or corner cases in the UX.

For transitions between activities navigation buttons are provided and navigation drawer panel.
Also when user taps the widget app will come to foreground.

Describe any libraries you'll be using and share your reasoning for including them.

Picasso or Glide to handle the loading and caching of images.
Room for data persistence.
LiveData and ViewModel are used for tracking the changes in the database and keeping UI in consistent state.
Retrofit for remote server interactions.
GSON or Moshi for parsing JSON into Java objects.
AppCompat for backward compatibility.
LIbraries from android.support.design package for implementing Material Design user interface recommendations.
Google Play services SDK for utilizing the AdMob and Analytics Google Play Services

Describe how you will implement Google Play Services or other external services.

Trivia Quiz will utilize Open Trivia DataBase https://opentdb.com for getting the quiz questions.
Also AdMob will be used - there will be an ad-banner placed on the "About" activity.
And the app will use mobile-app reporting system from Google - Google Analytics. Which will help to understand how people use Trivia Quiz.

Next Steps: Required Tasks

Task 1: Project Setup

- Update Android Studio and Gradle to stable versions
- Plug & Configure libraries
- Trivia Quiz will use pictures for avatars, so I'll have to get them =)

Task 2: Implement UI for Each Activity

- Build UI for UserSetupActivity
- Build UI for ChoosingCategoriesActivity
- Build UI for QuizSetupActivity
- Build UI for QuestionActivity with DrawerLayout
- Build UI for ScoreActivity with DrawerLayout

Task 3: Modeling and networking setup

- Create model classes
- Implement networking and JSON processing
- Handle all input error prone cases

Task 4: Persistence

- Implement persistence mechanisms in conjunction with LiveData and ModelView

Task 5: Google Play Services

- Implement Google Play Services (Analytics & AdMob)

General conditions:

App will be written solely in the Java Programming Language.
App will conform to common standards found in the Android Nanodegree General Project Guidelines.
App will utilize stable release versions (not beta) of all libraries, Gradle, and Android Studio.
App will validate all input from servers and users. If data does not exist or is in the wrong format, the app logs this fact and does not crash.
App will include support for accessibility. That includes content descriptions, navigation using a D-pad.
App will keep all strings in a strings.xml file and enables RTL layout switching on all layouts.
App will perform short duration, on-demand network requests (for getting data from the server) and for these purposes app will use an AsyncTask.

How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”