**GitHub Username**: gustavoballeste

# TBD (a trivia app)

## Description

Trivia (TBD) is a fun question and answer game of the most varied themes that you can play at any time and accumulate points every round.
Choose a category and challenge your knowledge.
The data will be provided by https://opentdb.com/

## Intended User

This app is for people who like to challenge their own knowledge in many different categories.

## Features

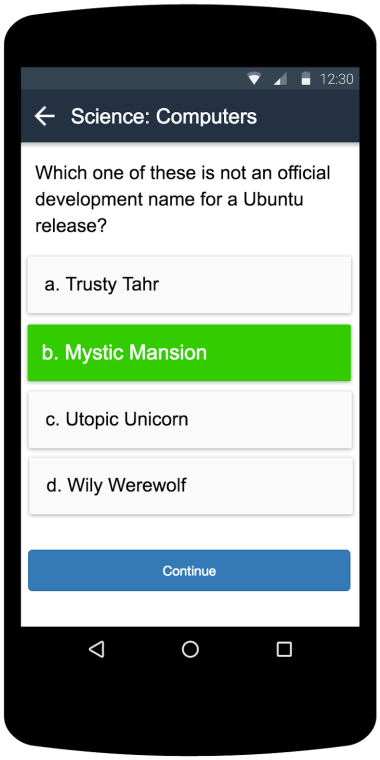
List the category types.
Ask questions, where the user should choose a response.
Display the quiz result.
Post user points.

# User Interface Mocks

## Screen 1 - Main (Exercise history)

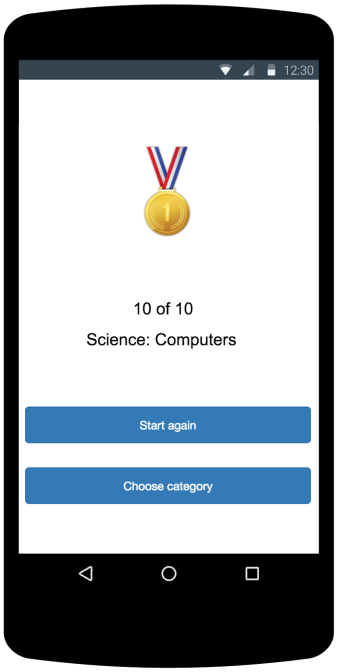

Displays the categories that can be chosen

## Screen 2 - Quiz (Detail)




Displays the question and possible answers.

## Screen 3 - Result




Displays the result of the round.

# Key Considerations

### How will your app handle data persistence?

A contract class will be built with the quiz information (category, type, question, correct answer, other answers, result) and also an SQLiteOpenHelper extension to create the database based on the contract information.
The endpoint data query https://opentdb.com/ will be started from an extension of the AsyncTask class, within the doInBackground method (Among some objects used, the main ones are HttpURLConnection, Uri, JSONObject, JSONArray, Vector <ContentValues>).
The result will be included within the database via ContentProvider, which will also have the function of changing, querying and deleting information from the database.
The data presentation will be via Fragment (Loader).

### Describe any corner cases in the UX.

From the home screen, the user can select a category.
From the quiz screen, the user chooses a response and after the result, he has the option to go to the next question.
From the quiz screen, the user has the option to go back to the home screen to choose another category, using the back button.
At the end of a round of questions, the score is displayed and the user has the option to start a new round or return to the home screen to select another category.

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

com.android.support:cardview-v7 - Will be used on the home screen
compile 'com.android.support:appcompat-v7 - App bar
com.jakewharton:butterknife - Automatically cast the corresponding view in the layout.
com.google.android.gms:play-services-ads - Banner and Interstitial
com.google.android.gms:play-services-analytics - Measure user activity

### Describe how you will implement Google Play Services.

Google Analytics - All user navigation
Google AdMob - A banner below both 2 screens and an interstitial at the end of a round of questions

# Next Steps: Required Tasks

## Task 1: Project Setup

- Configure libraries
- Define resource files (category icons)

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity (Grid)
- Build UI for Detail Activity (Question)

### Task 3: Build Contract

- Build contract
- Build DB Helper

### Task 4: Fetch and parse JSON data

- Create quiz list from JSON
- Load contract by AsyncTask

### Task 5: Show the data on View

- Show the questions on the screen
- Format the data
- Create the navigation between screens

### Task 6: Embed the tablet layout

- Draw tablet screen
- Adjust the code

### Task 7: Test the app

- Run all unit codes
- Run all functional tests