

Struktur

Die App besteht aus drei Hauptaktivitäten:

1. **StartQuizActivity** – Hauptoberfläche zur Auswahl eines Quiz.
2. **QuizActivity** – Logik zum Anzeigen von Fragen und Verarbeiten von Antworten.
3. **LastResultActivity** – Zeigt das letzte Quiz-Ergebnis an.

Die App verwendet:

- Buttons, TextViews, ProgressBar
- String-Ressourcen für Fragekategorien und Feedback
- Internen Speicher (quiz.txt) zum Speichern von Ergebnissen

1. StartQuizActivity.java

Zweck:

Benutzeroberfläche zur Auswahl eines Quiztyps und zum Starten des Quiz.

Wichtige Komponenten:

- btnPickType, btnBeginQuiz, btnSeeLast: UI-Buttons
- showPopupMenu(View v): Verwendet PopupMenu, um dem Benutzer eine Auswahl an Quizkategorien zu ermöglichen
- quizPicked: Speichert den ausgewählten Quiztyp aus den String-Ressourcen

Logik:

- Die Auswahl eines Quiz ermöglicht den Start-Button
- Startet QuizActivity mit dem ausgewählten Quiztyp als Intent-Extra
- Öffnet LastResultActivity, um das vorherige Ergebnis anzuzeigen

2. QuizActivity.java

Zweck:

Verarbeitet die Quiz-Logik: vom Laden der Fragen über Bewertung bis zum Speichern des Ergebnisses.

Wichtige Komponenten:

- TextViews: Für Frage-Nummer und Inhalt
- Button[] optionBtns: 4 Antwortmöglichkeiten je Frage
- ProgressBar loadingBar: Zeigt den Fortschritt der richtigen Antworten visuell an

- String[] quizQs, String[][] quizChoices, String[] quizAnswers: Datenarrays, geladen aus den Ressourcen

Logic Flow:

1. loadData() lädt die passenden Quizdaten basierend auf quizType
2. showNext() zeigt die nächste Frage und Antwortmöglichkeiten
3. checkAns() prüft die Antwort, gibt Feedback, deaktiviert die Auswahl und aktiviert den Weiter-Button
4. Verfolgt Punktestand und erkennt das Quizende
5. Bei Abschluss wird saveResult() aufgerufen, um das Ergebnis in quiz.txt zu schreiben

Speicherung:

- Verwendet FileOutputStream, um folgende Daten zu speichern:
 - Quiz-Typ
 - Anzahl richtiger und falscher Antworten

3. LastResultActivity.java

Zweck:

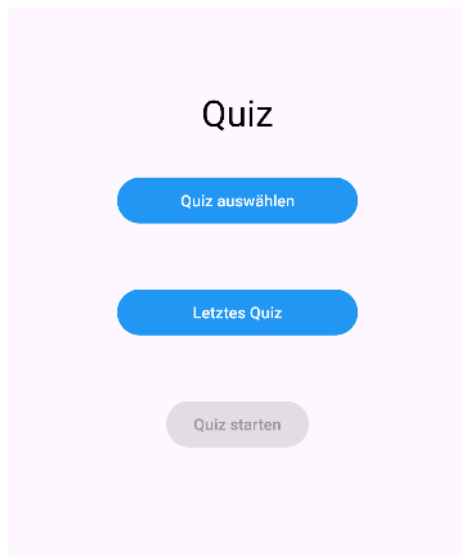
Liest das gespeicherte Quiz-Ergebnis aus quiz.txt und zeigt es an.

Wichtige Komponenten:

- TextViews zur Anzeige von:
 - Quiztyp
 - Richtige Antworten
 - Falsche Antworten

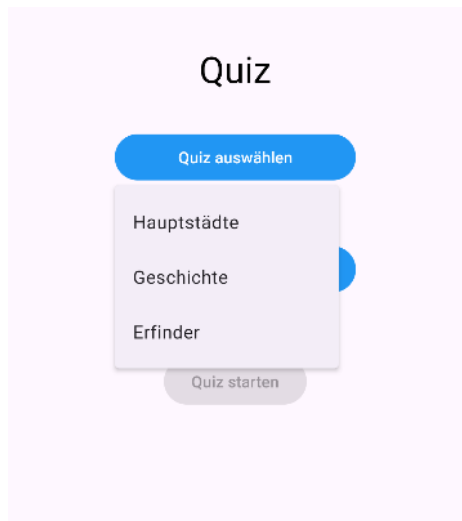
Logik:

- readData() liest aus der Datei im internen Speicher
- Extrahiert 3 Zeilen:
 - Quiz-Kategorie
 - Richtige Antworten
 - Falsche Antworten



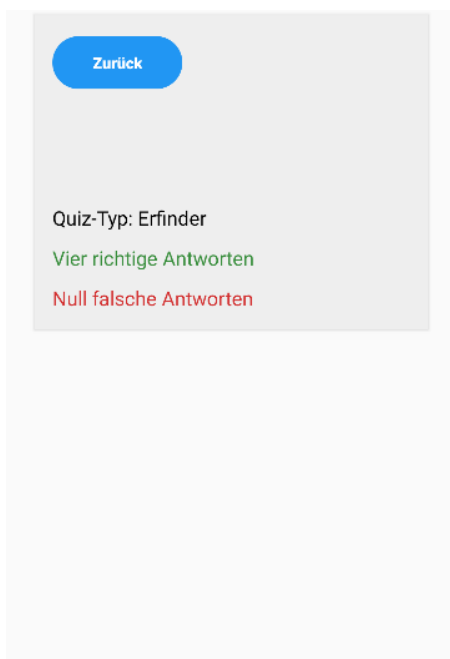
Hier sieht man die Startansicht der Quiz-App. In der Mitte befinden sich 3 Schaltflächen, die man auswählen kann. Momentan ist die dritte (in grau) nicht auswählbar.

- Erste – „Quiz auswählen“ – hiermit kann man das Quiztyp auswählen
- Zweite- „Letztes Quiz“ – damit kann man das Ergebnis des zuletzt absolvierten Quiz sehen
- Dritte – „Quiz starten“ – Man kann das Quiz anfangen, sobald das Quiztyp ausgewählt wird

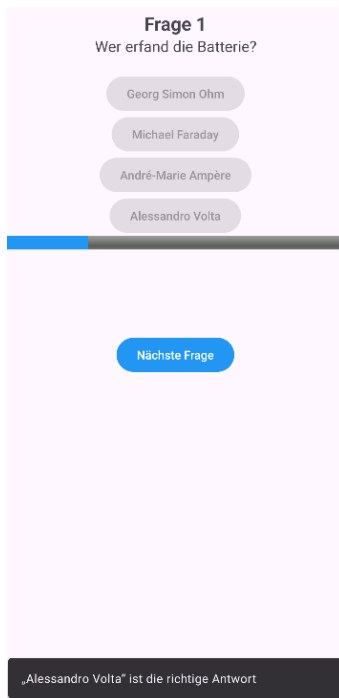


Nach dem Klick auf die Schaltfläche „Quiz auswählen“, kann man eine der drei Quizkategorien wählen:

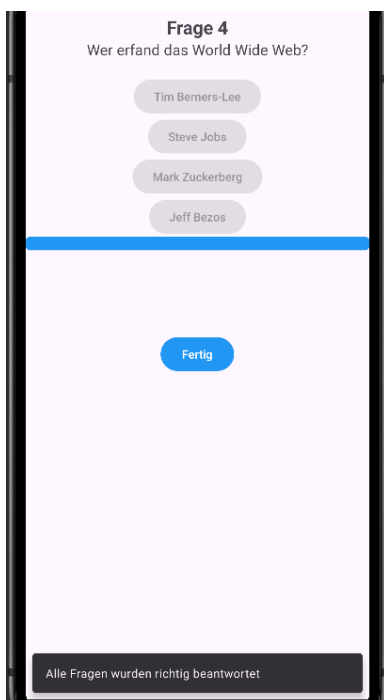
- Hauptstädte
- Geschichte
- Erfinder



Nach dem Abschließen von einer Quiz wird das Ergebnis gezeigt. Dabei kann man das Quiz-typ, als auch die Anzahl von richtigen und falschen Antworten sehen. Mit der Schaltfläche „Zurück“ (oben links) kann man zurück auf die Startseite landen.



In diesem Bild sieht man die erste Frage der Quiz. Die Antworten sind eingeblendet, nachdem man eine Option auswählt. Unten sieht man die richtige Antwort, als auch der Fortschritt im Quiz. Momentan kann man auf die nächste Frage gehen.



Hier sieht man die letzte Frage in dem Quiz. Die Fortschrittbalken zeigt, dass die letzte Frage beantwortet ist. Am unteren Bildschirmrand wird in einer eingeblendeten Nachricht bestätigt: „Alle Fragen wurden richtig beantwortet“.

In der Mitte befindet sich die Schaltfläche „Fertig“, um das Quiz abzuschließen. Dies deutet darauf hin, dass der Benutzer das gesamte Quiz korrekt gelöst hat.

Selbstständigkeitserklärung:

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte Hilfe angefertigt habe