“Schnitzeljagd”

Technik mobiler Systeme (B.54) - SoSe15

Gliederung

- Anforderungen
- Umsetzung
- Testprotokollauszug
- Auswertung

Anforderungen

analoge Schnitzeljagd

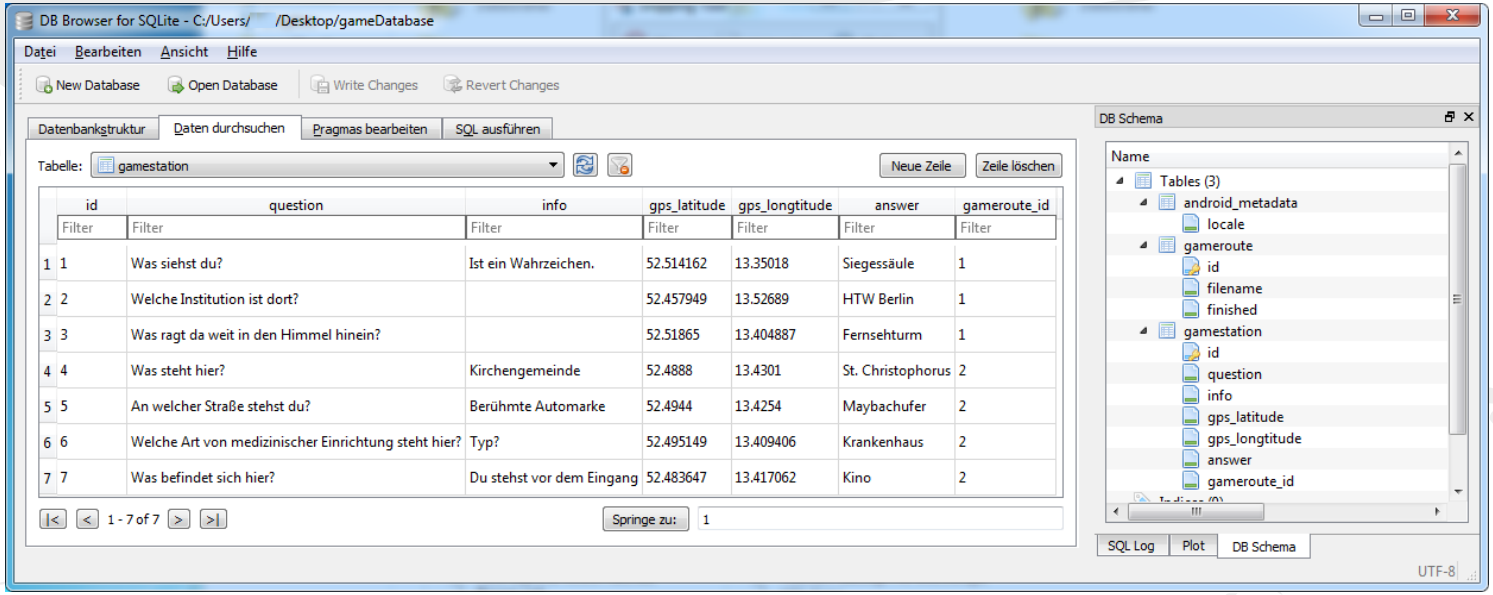
- Spielersteller / Spieler
- eigenständiges navigieren
- ortsgebundene Fragen / Hinweise
- Ziel: Lösen aller Stationen

Projekt “Schnitzeljagd”

- Android App realisiert das analoge Spielprinzip
- Unterstützung durch
  - GPS-Koordinaten
  - mobile Internetverbindung
- Prüfung übernimmt ein System
- verwalten verschiedener Spiele

Umsetzung

- Datenbankstruktur mit DB Browser for SQLite



Umsetzung

- SQLite Datenbank

```
public List<List<String>> getAllGameRoutesWithoutGameSationsFromDatabase() {
   List<List<String>> gameRouteList = new ArrayList<List<String>>();
   String selectQuery = "SELECT  * FROM " + TABLE_GameRoute;
   SQLiteDatabase db = this.getReadableDatabase();
   Cursor c = db.rawQuery(selectQuery, null);
   if (c.moveToFirst()) {
       do {
           Integer gameID = c.getInt((c.getColumnIndex(GameRoute_ID)));
           String filename = c.getString((c.getColumnIndex(GameRoute_Filename)));
           Boolean finished = c.getInt((c.getColumnIndex(GameRoute_Finished))) > 0;
           List<String> tempList = new ArrayList<String>();
           tempList.add(gameID.toString());
           tempList.add(filename);
           tempList.add(finished.toString());
           gameRouteList.add(tempList);
       } while (c.moveToNext());
   }return gameRouteList;
```

Umsetzung

- SupportMapFragment

```
GoogleMap googleMap = ((SupportMapFragment)getSupportFragmentManager().findFragmentById(R.id.mapFragment)).getMap();
googleMap.setMyLocationEnabled(true);
googleMap.getUiSettings().setZoomControlsEnabled(true);

double latitude = currentGameStation.getGeoPoint().getLatitude();
double longitude = currentGameStation.getGeoPoint().getLongitude();
googleMap.addMarker(new MarkerOptions().position(new LatLng(latitude, longitude)).title("Schnitzelpoint"));
CameraPosition cameraPosition = new CameraPosition.Builder()
                .target(new LatLng(latitude, longitude))
                .zoom(15.0f)
                .bearing(0.0f)
                .tilt(45.0f)
                .build();
googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));
```

Umsetzung

- Beispiel: Kreuzkölln.schnitzel


```
{
  "gameroutes": [
    {
      "answer": "St. Christophorus",
      "srid": 4326,
      "number": 1,
      "longitude": 13.4301,
      "latitude": 52.4888,
      "question": "Was steht hier?",
      "info": "Kirchengemeinde"
    },
    {
      "answer": "Maybachufer",
      "srid": 4326,
      "number": 2,
      "longitude": 13.4254,
      "latitude": 52.4944,
      "question": "An welcher Straße stehst du?",
      "info": "Berühmte Automarke"
    },
    {
      "answer": "Krankenhaus",
      "srid": 4326,
      "number": 3,
      "longitude": 13.4094,
      "latitude": 52.4951,
      "question": "Welche Art von medizinischer Einrichtung steht hier?",
      "info": "Typ?"
    },
    {
      "answer": "Kino",
      "srid": 4326,
      "number": 4,
      "longitude": 13.4171,
      "latitude": 52.4836,
      "question": "Was befindet sich hier?",
      "info": "Du stehst vor dem Eingang"
    }
  ],
  "number of gameroutes": 4,
  "version": 1
}
```

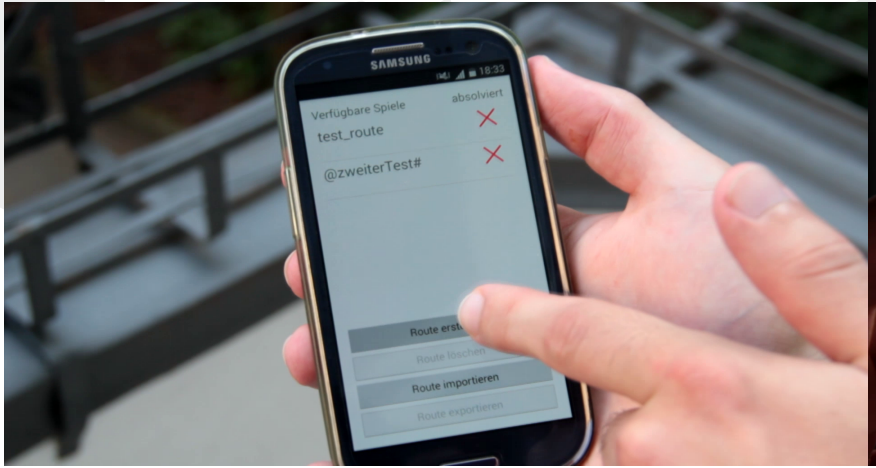
Umsetzung

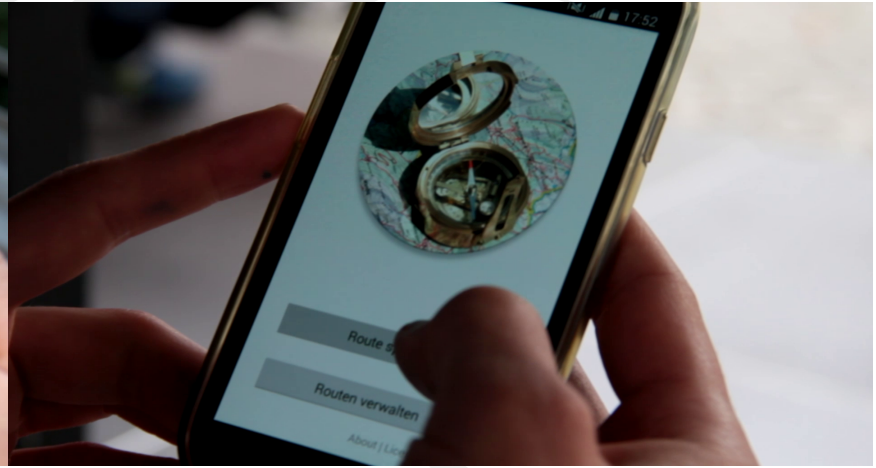
- Import

```
String jsonDataFromFile = IOHelper.readText(file.getAbsolutePath());
JSONObject jsonRoute = JsonHelper.parse(jsonDataFromFile);
GameRoute gameRoute = JsonHelper.toGameRoute(jsonRoute, file.getName());
GameManager.insertNewGameRouteWithGameStaionsInDatabase(gameroute);
```

- Export

```
String jsonGameRoute = JsonHelper.toJson(gameRoute).toString();
String filename = gameRoute.getName() + ".schnitzel";
IOHelper.writeText(jsonGameRoute, filename, sfManager.getSchnitzelFilesDir());
```





Video & live Demo

Testprotokollauszug

Bezeichnung	Beschreibung	Erwartung	Ergebnis
Feature 1 (M)	Anzeigen einer Spielstation innerhalb eines Spiels.	Es werden Stationsnummer, zu beantwortende Frage, Hinweisbutton, Kartenausschnitt mit Route von aktueller Position bis Stationsposition (Ziel), Eingabefeld für Antwort, ein Button der die Rückkehr zum App-Hauptmenü ermöglicht sowie ein Button der die Überprüfung der eingegebenen Antwort ermöglicht angezeigt.	alles vorhanden, erfolgreich getestet und durch Testpersonen bestätigt
Feature 2 (M)	Es wird die aktuelle Position auf einem Kartenausschnitt der Spielstationsseite angezeigt.	Auf dem Kartenausschnitt ist die aktuelle Position + / - 10 m korrekt als Punkt dargestellt.	Erfolgreich auf ~3-5m Genauigkeit getestet
Feature 3 (M)	Es wird ein zusätzlicher Hinweistext für eine Spielstation hinterlegt.	Nach Klick auf den Hinweisbutton der Spielstationsseite erscheint eine Einblendung mit dem zusätzlich hinterlegten Hinweistext.	Vorhanden, erfolgreich getestet
Feature 4 (M)	Eine Spielstation wurde erfolgreich absolviert.	Die eingegebene Antwort ist identisch mit der hinterlegten Lösung. Es wird die nächste Spielstation geladen, existiert keine weitere Spielstation ist die Spielroute erfolgreich absolviert.	kein case sensitiv nötig, aber identische Antwort erforderlich; erfolgreich getestet – siehe Video
Feature 5 (M)	Eine Spielroute wurde erfolgreich absolviert.	Die letzte Spielstation wurde erfolgreich absolviert. Es erscheint eine entsprechende Einblendung mit Erfolgsmeldung und eine Statuskennzeichnung in der Auswahlliste der verfügbaren Spielrouten.	erfolgreich getestet – siehe Videobeweis
Feature 6 (M)	Es wird eine gültige Spielroute-Datei importiert.	Einblendung einer entsprechenden Erfolgsmeldung, die importierte Spielroute ist in der DB hinterlegt und in der Liste der verfügbaren Spielrouten mit „nicht absolviert"-Symbol als Statuskennzeichnung sichtbar.	Vorhanden und getestet
Feature 7 (M)	Anzeige aller verfügbaren Spielrouten.	In einer Auswahlliste (angezeigt unter „Route verwalten" oder „Route jetzt spielen") werden alle verfügbaren Spielrouten, die in der DB hinterlegt sind, mit Dateinamen sowie Spielstatus angezeigt.	Funktioniert, siehe Video

Auswertung

- Nachbesserung von fehlgeschlagenen Fail-Tests
- Offene Features realisieren
  - Route bearbeiten
  - Unterstützung von Landscape-Modus
  - Refactoring und Qualitätssicherung
- professionelles Layout entwickeln

- Veröffentlichung auf GitHub

Vielen Dank
für Ihre
Aufmerksamkeit