
GitHub Username: bjoerngam

StolperSteineAR

Description

In Europe especially in Germany we're having an art project: Stolpersteine (stumbling stones). You can find this stones nearly all around Europe and this stones are a cenotaph for the victims of the Nazis, which has been deported to the concentration camps.

You can find detailed informations here:
https://en.wikipedia.org/wiki/Stolperstein
https://www.youtube.com/watch?v=MixO_amdabs (from minute: 3:30 on)

Example of one "Stolperstein" in Hamburg, Germany:

I want to make sure with my AR app every interested person is been able to see the historical information about the victim as far as it's known. The idea is it to use AR for this project. You're a standing in front of a house and than you will see the information.

Intended User

Every interested person. Maybe you can use the App perfectly for students or for visitors.

Features

The user is getting information via the camera. It's a AR experience.

- AR experience
- Local DB will be used for storing the favourite information
- Via GPS the correct JSON information will be used.
- All Stolperstein informationen will be available via JSON. This file has to been stored online. Currently there is no webservice available or a plain JSON file.

User Interface Mocks

Main Screen

The view of the user of a Stolperstein.

Main Screen with AR (first page)

The app added the information into the camera display. And shows first the image of the victim. The first and the last name, and the date of birth and the date of death.

Main Screen with AR (second page)

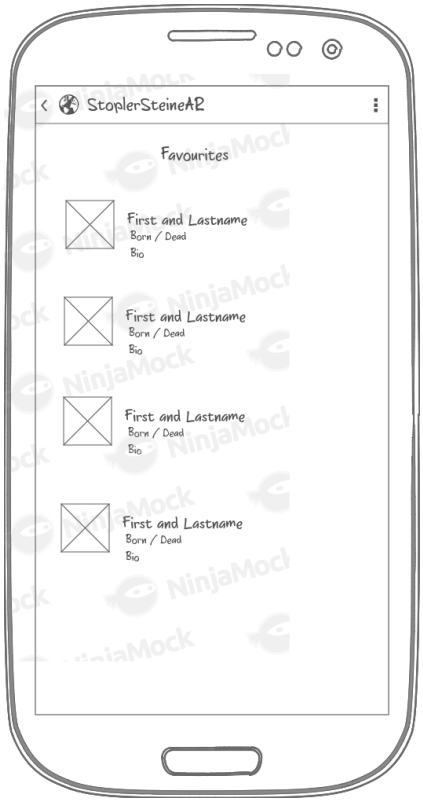
When the user clicks the AR information from screen 2 he will see biography of the victim. He can click the white circle for saving the information locally.

Main Screen Menu

In the menu the user has the opportunity to see the local stored favourites and also to open the About dialog. Here will also the information about the used libraries be shown.

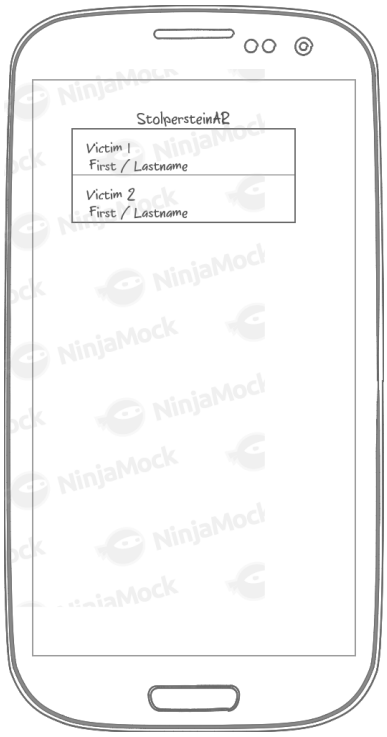
For the “Udacity-Testing Version” I will add a menu entry with some pre defined GPS coordinates.

Favourites Overview



Here can the user see his locally saved favourites.

Widget Screen



The Home-Screen Widget. With this the user has a quick access to his favourites (I have to find another word for this). If the user clicks one of the elements a “single view screen” will be displayed (screen 7).

Single Favourite

Key Considerations

How will your app handle data persistence?

All of the favourites will be stored locally via a Content Provider.

Describe any edge or corner cases in the UX.

There is always the “back arrow” in the upper left corner.

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

For parsing the JSON objects (GSON) and establish the HTTP connection: OKHTTP / Volley.
For reducing the complexity of the code. Butterknife and maybe Picasso for scaling the images.

For the whole AR topic I will use: Kudan. Of if this is getting to complicated: I will use the approach from: https://www.netguru.co/blog/augmented-reality-mobile-android

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

For getting the correct position I will use the Google fused location provider.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

In general I want to use a separate library for all of the dialogs and permissions dialogs.

The project tasks:

1. I have to convert the CSV file (https://www.netzwolf.info/mole/) into a JSON file and place it into the internet maybe on my private page or into Firebase.
2. I will create everything around getting the JSON information and make a HTTP connection.
3. I will create the MainActivity UI, the menu, the information dialog and the favourites UI.
4. I will create the Google location provider.
5. I will create the stolperstein UI.
6. I will create everything around the AR topic.
7. I will create the database and content provider.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for ARActiivity
- Build UI for FavouritesActivity

Task 3: Implement the JSON logic, getting the JSON data via an IntentService and implementing a Loader for the JSON data

Convert the CSV file into a nice JSON structure and place this file online.
Developing the whole JSON / HTTP logic.

- Convert the current file via script.
- Check how to place the file into firebase / or into my page
- Currently there is no webservice available for the stolperstein data. So I will import the plain CSV file into Firebase transformed it into a JSON file.
- (https://stackoverflow.com/questions/33930915/how-to-import-csv-files-into-firebase/33932989)
- Creating HTTP logic and check if we're getting a working HTTP connection.
- Parsing the JSON objects.

- I will use a LoaderClass for loading the JSON class data into the different views and it will be necessary and it would be wise to use parcelable for parsing the different information from one intent to another.

Task 4: The dialogs library

Creating the basic dialogs, permissions dialogs in a separate library. And creating a basic MainActivity UI.

- Dialog: No working internet connection
- Dialog: JSON parsing error
- Dialog: Database error
- Permission dialog: Location service
- Permission dialog: Camera

Task 5: The Google Location service

Developing the Google location service. Getting the current position and save this postion as a sharedPrefs.

Task 6: The UI for one stolperstein

I will create the UI for one stolperstein.

The UI contains:

- Image (if available)
- Birthdate and date of death
- On the backside:
    - Biography
    - And a circle for storing the information locally.

Task 7: The AR task

Setting up everything around the AR. With the position from task: 5 I want to get the correct JSON object. I have of course to add a small tolerance for the latitude and longitude. If I'm having a match I will display the information of the current object. We will find at least one stolpertstein here.

- Either using the library or the approach.

Task 8: Content Provider and database

Creating the content Provider for the local database and adding the ability to store the data locally.

Task 9: Udacity Test Menu item

Creating some dummy GPS data for testing purpose and also creating a second Gradle configuration.

Task 10: Home Screen Widget

Creating the Home Screen Widget.

- Creating the UI of the Widget
- Creating the UI for the SingleFavouriteView
- Develop the Widget

Task 11: After the positive review

Publish the app in the Google Play Store.
