THE WEATHER GAME

1. Introduction

I have developed an app that puts a twist on the standard weather app. I love geography and learning about climates in different regions. Therefore, I often see if I can guess the temperature of random cities in the world. I decided to make an app that would simulate this game. The app consists of two main sections. The first section is a weather section – in this section, the user can find the weather in different cities, by typing the city name in. The second section is the ‘play’ section. In this section, the user will be shown a random city in the world. The user will then have to guess what the current temperature in that city is. Furthermore, if the user guesses the temperature correctly, they earn a point.

Features of the app

* Find weather of cities
* See the forecast of the city for the next 5 days
* Play a weather game and earn points

2. Description of key app components

I have used activities in my app that extend AppCompatActivity.class. This was done to enable UI interaction.

The main activity of the app launches when opened. The main activity uses buttons through which the user can redirect themselves to the different pages of the application. The main activity also incorporates a menu. The menu has a contact, about and points item. These items can be used to learn more about the app. The points item launches an activity which displays how many points the user has won. There is also an option to reset the points value, if the user wants to start the game again from scratch.

The user can also go to the ‘play’ section. Here, the user can click on a number of buttons that interact with the user. The user can click on the ‘new city’ button to be shown the name of a new city for which they will have to guess the temperature. The user can then enter the value in an EditText box and then click on a submit button. They will then be shown a toast telling them if their answer was right or wrong. If the answer is right they get one point. If they don’t know the answer they can click on ‘get answer’ button, which will show them the correct answer.

If the user gets intrigued by the city, and wants to learn more about the city, they can simply click on the displayed city name, and they will be redirected to a google search of the city.

The other section is the weather section. Here, the user can learn about the weather in different cities. The app shows the current temperature and condition in the city queried. It also shows detailed weather forecasts for 5 days. If the user wants to research more about the weather in the city, if they click on the weather icon, they will be redirected to a google search of the weather in that city so they explore more about it.

3. Design rationale

I have used a number of different android features. I have used Recycler View to display the list of weather forecasts of the next few days. This list is scrollable. I had initially implemented it with list view, but since recycler view is more efficient, I decided to use recycler view. As I used recycler view instead of the list view, I was required to do some extra work in the implementation – like binding the data to the view holder. I used an apdater to provide the date to the recycler view.

I also used the Singleton factory pattern for making API requests. I used multiple methods to make different API requests. When a use enters a city, I had to find the city id of the city so I could query the API, as the API didn’t accept city names. I then had to find the weather according to the id. Since there were so many different request originating from multiple areas of the app, I decided to make the singleton pattern so there would only be one request queue.

I also used the volley library for the API requests. It was crucial that the networking tasks should not occur in the UI thread, as they take long periods of time. Therefore, networking tasks should be performed in an asynchronous manner. The volley library performs this by itself. The library is quick for networking tasks and so I decided to use it. It uses a separate thread for the networking tasks making the app faster.

I also used the Picasso library for getting images from a url. To display the weather appropriate images, Picasso is very useful.

I also have used shared preferences. I used this to store the points earned by the user. Since it is a very small integer value, it is appropriate for storing through shared preferences. Shared preferences also offer persistent storage which is required for this feature.

The app also uses the internet for the data retrieval, however, when the app is not connected to the internet it can still launch and display screens.

An important design decision I made is the incorporation of rounded corners. These are heavily used in modern app design. Therefore, I created a standard drawable shape – card.xml. I used this as the background for many buttons and views so that the elements would have rounded corners and look modern.

I also used linear layouts and relative layouts heavily. I used linear layouts so that elements could be placed after each other without overlap. I also used layouts inside layouts so I could position elements in a manner that looked good.

I also used an empty view so I could have a horizontal line though the screen,

I also used text views, edit views and buttons.

I also used a random number function so that I could generate a random city to display.

4. Reflection on the development

I took a structured approach in development. I first created designs and wireframes to understand what would look good. I then listed out all the features I wanted the app to have and divided them into must haves, could haves, should haves, and shouldn’t haves. This gave me a lot of clarity. I then decided which classes were required and which API to use. I then began programming it and continued testing simultaneously. The good practices were testing and designing the app thoroughly to ensure there would be no crashes.

5. How to improve the app in future

The app can be improved in the future by making accounts so you can keep track of your score across devices. I can also make leader boards so friends can compare scores. I can also allow users to guess temperature, wind and max and min temperatures.

6.References

[*https://www.metaweather.com/*](https://www.metaweather.com/)*,* Meta weather

www.youtube.com. (n.d.). *Java Android App using REST API - Network Data in Android Course*. [online] Available at: https://www.youtube.com/watch?v=xPi-z3nOcn8&ab\_channel=freeCodeCamp.org [Accessed 17 Mar. 2022].

Volley. (n.d.). *Volley overview*. [online] Available at: https://google.github.io/volley/ [Accessed 17 Mar. 2022].

Android Developers. (2019). *Build your first app  |  Android Developers*. [online] Available at: https://developer.android.com/training/basics/firstapp.