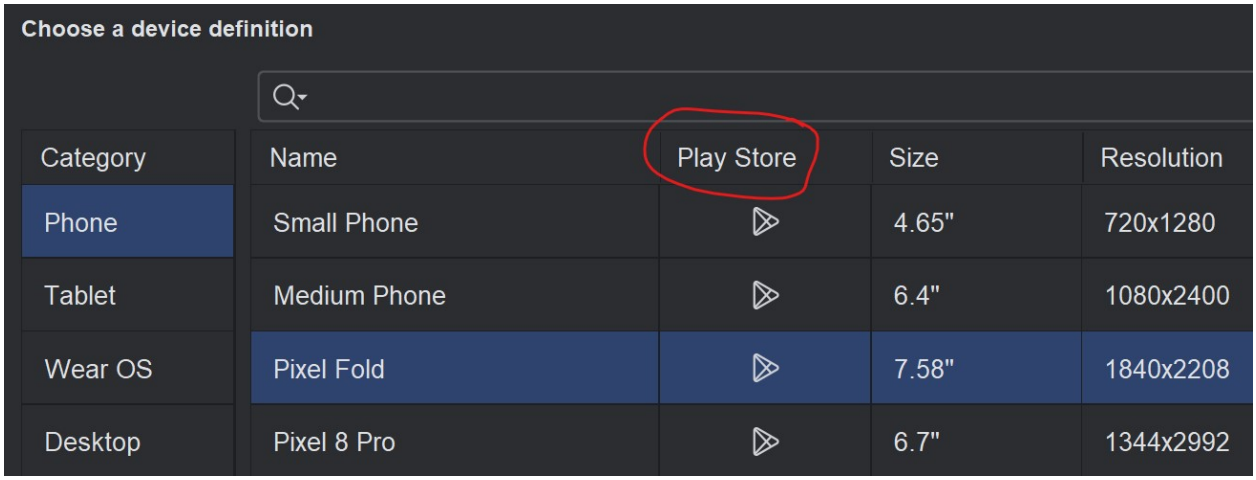
Android programming: Flipped Classroom: Maps

In order to use Maps in your app, you must have a device or emulator with Google Play Serives enabled. When you create your phone, most of them have "Play Store" enabled. This is what it looks like.



You will need the Android SDK for Google Maps. Your gradle build file has the proper dependence, so no problems there. You should visit this console site (`https://console.cloud.google.com/`) to make sure you have an account and then you should create an API key. Restrict your API key to Android Apps, especially the one with your package name and signing key. Then you need to enable some APIs, like the Maps SDK for Android and the Geocoding API.

Here is a demo video `https://www.youtube.com/watch?v=TbGOmD1BOjc`. Here are some instructions for getting a map API key. `https://developers.google.com/maps/documentation/embed/get-api-key`. Google has great documentation and I show some pointers in the video. I also provide a screen picture of a properly provisioned API key at the end of this document.

The Maps APIs displays maps and the Geocoder API translates human-readable addresses (even partial and misspelled ones) into Latitude and Longitude locations.

As always, look for `XXX Write me` in the code.

Rules and submission Recall the rules for flipped classrooms: do the work alone or with a partner that is unique for the semester. You can find the github link in canvas or on piazza.

Only one student needs to submit this code, but both students may do so if they wish. This assignment should be submitted through github (sorry, partners can't share the repository). Include a README at the top directory level that contains both student's EIDs.

- **AndroidManifest.xml** Add `ACCESS_FINE_LOCATION` permission.
- **values/google_maps_api.xml** Put your API key here.
- **activity_main.xml** You should be fine with this.
- **content_main.xml** You need to replicate the layout in the video, which has an EditText followed by two buttons, one with a picture of an X in it (`ic_clear_black_24dp.xml`), the other with the word "Go" in it. The EditText should specify `android:imeOptions="actionDone"` which means that if you hit the return key on the soft keyboard, it lets our code know that it is time to geocode and move the map. You should also specify `android:maxLines=1` and `android:inputType="textPostalAddress"`.

  There is code in MainActivity that assumes the EditText object is called `mapET` and the go button is called `goBut`. You can use these names or change the code to whatever names you use.

  You will also have to place the map fragment in the layout.
- **MainActivity.kt** In `onCreate` you need to get a reference to the map, then initialize it (via `getMapAsync`). You also need to initialize the Geocoder object.
  - When you push the "Go" button, you should read the EditText box and geocode that address to get a lat/long pair. If the geocoding succeeds, use the first suggested address and move the map's camera to that location at zoom level 15.0f.
  - **onMapReady** If we have location permissions, then enable the "my location" function on the map (i.e., the blue dot) and enable the "find my location" button on the map. To test this functionality, you should set your emulated phone to have a location in the bounds of the map when it starts. If you are using a real device, testing achieved!
  - The clear button should clear all markers from the map. So should a long click on the map itself.
  - Clicking the map should create a marker whose title is the lattitude and longitude but ONLY to 3 decimal places of precision.
  - Find the location of the Harry Ransom center and start the map at that location, zoom factor 15.0.

Google Cloud | FCGoogleMaps | Sea

API APIs & Services

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Page usage agreements

Name *
API key 1

Key restrictions

Restricting where and for which APIs this key can be used helps prevent unauthorized use. Learn more

Set an application restriction

Application restrictions limit an API key's usage to specific websites, IP addresses, Android applications, or iOS applications. You can set one application restriction per key.

- ○ None
- ○ Websites
- ○ IP addresses
- ◉ Android apps
- ○ iOS apps

Android restrictions

Restricts API key usage to specified Android apps. Add the package name and SHA-1 certificate fingerprint for each app.

ADD

Filter Enter property name or value

Status	Package name	Fingerprint	Edit
	edu.cs371m.fcgooglemaps	1D:C3:57:95:D5:20:A0:43:6C:E	
	edu.cs371m.fcgooglemaps	A5:56:AC:05:8E:59:10:D2:14:1	
	edu.cs371m.fcgooglemaps	D0:A2:AE:C8:F0:9B:A7:04:8C:D	

API restrictions

API restrictions specify the enabled APIs that this key can call

- ○ Don't restrict key
  This key can call any API
- ◉ Restrict key

2 APIs

Selected APIs:

Maps SDK for Android

Geocoding API