Developing a Mobile Mapping App for Android Devices

Eric (Yeu-jer) Cheng

Master of Science in Geographic Information Science (MSGISci)

Department of Geography, California State University, Long Beach

Introduction

The rapid adoption of mobile computing involving smartphones and tablet computers has made it possible for millions of people all over the world to access and share geospatial data through mobile apps. For this project, I wanted to learn how to create a mobile mapping app that could take advantage of modern mobile device functionality including touchscreen interactivity, data access over the internet, and geolocation services.

The original purpose of the app was to fulfill the goals of the Calliope project – a cartographic design project begun by multiple departments on the California State University, Long Beach campus to communicate geospatial data through sound.

I chose to develop an app for the Android operating system because it is Linux-based, open source, and has the largest user base of any mobile operating system. The app was developed using the ArcGIS Runtime SDK for Android because the CSULB Geography department had multiple map and image services already published with ArcGIS for Server which could be accessed by the app.

The project does not have a study area in the traditional sense, but the mobile app does have its map's default extent centred on the city of Long Beach.

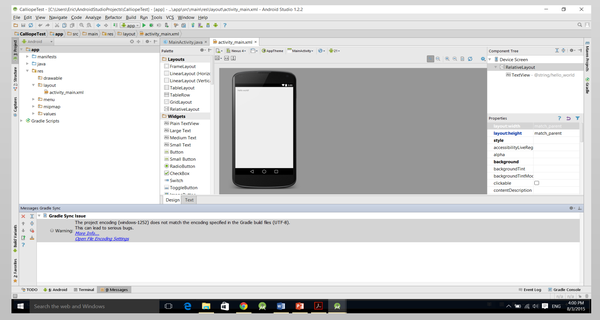

Figure 1. *Android Studio interface.*

Data and Data Sources

This project did not require much data acquisition, processing, and management because the CSULB Geography department had already created the map and image services that would be incorporated into the app prior to the start of this project. The DEM of the City of Long Beach was acquired through the Los Angeles Region Imagery Acquisition Consortium (LARIAC). The ArcGIS basemap was loaded directly from ArcGIS.com.

Table 1. *List of data and data sources used in the project*

Dataset	Source
Long Beach DEM (pixelated)	LARIAC1 (2006) via CSULB
Long Beach DEM (interpolated)	LARIAC1 (2006) via CSULB
ArcGIS Basemap	ArcGIS.com

Methodology

Android Studio was downloaded and installed to build the mobile app, and API 15: Android 4.0.3 (IceCreamSandwich) was selected for the app.

The first step was to create an app and load the map from Esri on it. Runtime Android SDK dependency would be downloaded and added from Esri ArcGIS Maven central repository after the app was built and run. The dependencies and packaging options were added to the Android Archive (AAR) package to the app module in build.gradle (Module:app). In main_activity.xml, Esri map characteristics were modified and the CSULB campus was set to be the center by modifying the map center point coordinates.

The next step was to add the DEM as a dynamic map service layer by adding the URL for the image service into String.xml.

The last step was to retrieve the coordinates where the user tapped the screen. An single tap listener was established and a callout to display the coordinates in the mobile app was created.

```
public class MainActivity extends ActionBarActivity {
    MapView mMapView = null;
    ArcGISDynamicMapServiceLayer mDynamicMapServiceLayer;
    GraphicsLayer mGraphicsLayer;
    boolean mIsMapLoaded;
    String mDynamicServiceURL;
    Point mapPoint;
    Callout mCallout;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Retrieve the map and initial extent from XML layout
        mMapView = (MapView) findViewById(R.id.map);
        // Get the dynamic service URL from values->strings.xml
        mDynamicServiceURL = this.getResources().getString(R.string.dynamicServiceURL);
        // Add dynamic layer to the MapView
        mDynamicMapServiceLayer = new ArcGISDynamicMapServiceLayer(mDynamicServiceURL);
        mMapView.addLayer(mDynamicMapServiceLayer);
        // Add Graphics layer to the MapView
        mGraphicsLayer = new GraphicsLayer();
        mMapView.addLayer(mGraphicsLayer);
```

Figure 2. Variable declaration and addition of map layers.

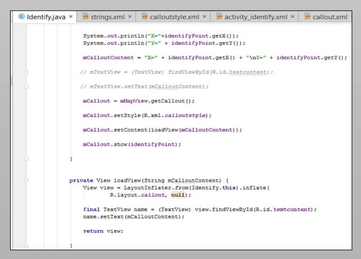

Figure 3. Callout window setup.

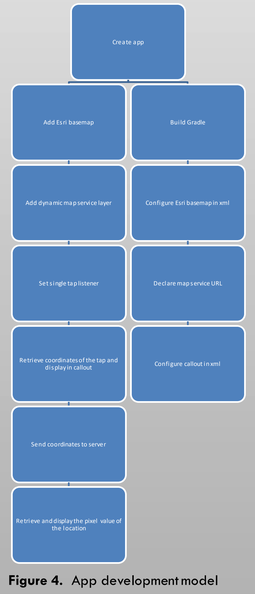

Figure 4. App development model

Timeline

Start Date	End Date	Job Functions
June 01, 2015	June 23, 2015	Load Basemap on Mobile App
June 24, 2015	June 24, 2015	Project Proposal Presentation
June 25, 2015	July 15, 2015	User Interaction
July 15, 2015	July 31, 2015	Callout with Coordinates
July 31, 2015	August 12, 2015	Project Report
August 12, 2015	August 14, 2015	Project Presentation and Poster
August 15, 2015	August 15 2015	Final Project Presentation

Results

The app can be successfully downloaded on an Android smartphone by connecting the phone to a computer with the app source files, and pushing the app to the phone with Android Studio. The app displays the Esri basemap centered on the CSULB campus with the predefined zoom level and map type. Then, a dynamic map service layer presenting the DEM of the city of Long Beach was added to the app. The zoom level and center were changed to view the entire DEM. Finally, the user can interact with the app by tapping on the screen. A callout window pops up with the geographical coordinates of the tapped location in it.

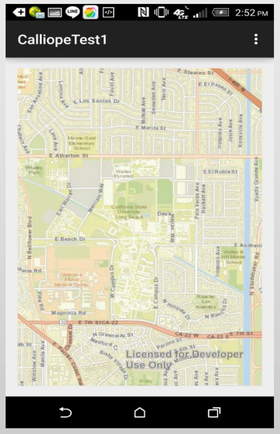

Figure 5. Esri basemap shown in mobile app.

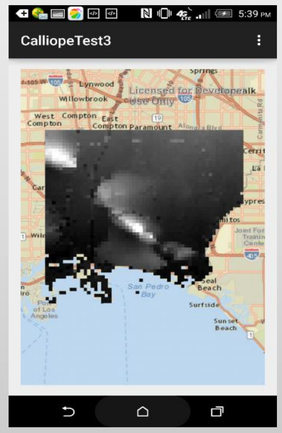

Figure 6. DEM shown in mobile app.

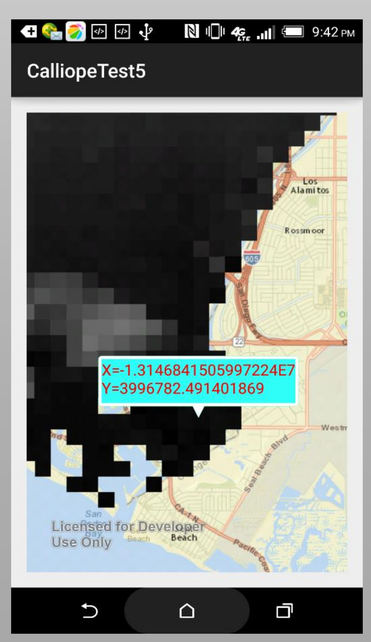

Figure 7. Callout displays the coordinates tapping occurred.

Discussion

Comparing to Eclipse, Android Studio was much more efficient for developing mobile app. However, Android Studio also required much higher computing powers.

Even though there were plenty of tutorials, documentations, and sample codes online, those resources could be outdated or erroneous. Also, sometimes it could be difficult to find information specifically for ArcGIS Runtime SDK.

For the app, each consequent step was taking the results from the pervious step as the foundation to be built upon. Therefore, whenever a delay occurred, the whole project was on halt. The most critical interruption in the development was to properly display the callout window. After the app recognized the user's tap, it obtained the coordinates of the tapped location, and created a callout. However, the callout did not have any content. Eventually, a text view was created to store the coordinates and display them inside the callout window. Then, I attempted to proceed with the project to send the coordinates to the server, and the corresponding pixel value would be returned. Identify task method was implemented, but the app failed to acquire pixel value from the map service. Identify task method was assumed to work in the similar fashion as the web app version, but it was not.

When the Calliope mobile app is eventually completed in the future, there are few ethical issues to be considered. First, the app will be locating the user, and this data gathering functionality must be consented to avoid privacy invasion. Also, what kind of sound to be used to represent different attributes can be challenging. For example, using gun fire sound to indicate areas with high crime rates is not appropriate.

Conclusion

The mobile app was still in development phase; however, this project provided the foundation for continuous development of the Calliope project or any mobile app using ArcGIS Runtime SDK. The pros and cons were compared for Eclipse and Android Studio, and Android Studio was the ideal IDE for mobile app development. The basics of loading and displaying maps in mobile app from ArcGIS for Server were learned. The procedures of interacting with user and presenting information in callout window were studied. The mobile app proved that it was possible to learn to create a mobile mapping app using online resources. However, the developer should be aware that not all resources online were correct and up to date. The mobile app's next step would be sending the coordinates to the map service and receiving the pixel value back. Then, the pixel value will be used to produce sound. The change in pixel values will lead to different pitches of the sound. More attributes can be added to be communicated with different sounds.

Submitted in partial fulfillment of the requirements of the Masters of Science in Geographic Information Science(MSGISci), August 15, 2015.

For additional information please contact:

Eric (Yeu-jer) Cheng at yeujer@gmail.com