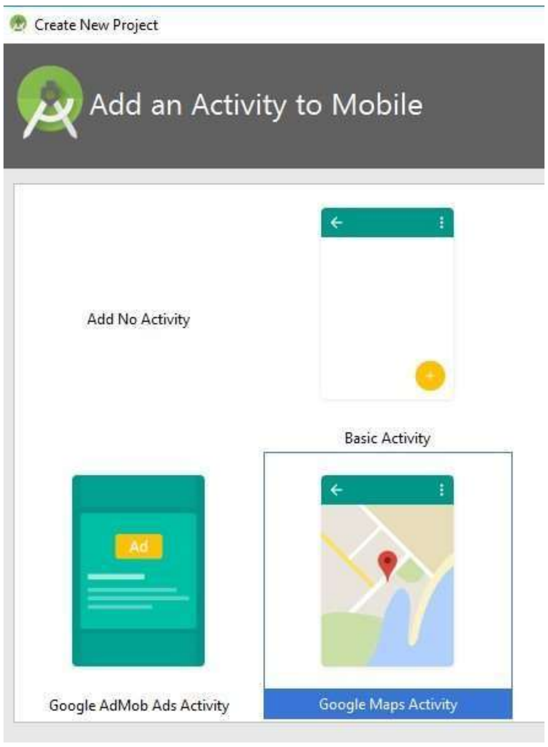
CSE2MAD Lab Week 8 – Part A Google Map Activity – Trace my steps (updated)

Aim: To learn how Google Map Fragments can be used in activities and how we can use location detection to update and overlay a map with information.

1) Create a new project with a "Google Maps Activity"



2) In the Android Manifest request permission for Fine and Coarse Location

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

3) Go to the res/values folder and open the google_maps_api.xml file

   Follow the instructions to obtain and insert an API key.

4) Go to the java file for the activity and fill out the class declaration

By default we extend FragmentActivity and we also have to implement OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, OnConnectionFailedListener

5) Let's declare some members

6)
```
private GoogleMap mMap;
 private FusedLocationProviderClient mFusedLocationClient;
 private LocationCallback locationCallback;
 private LocationRequest mLocationRequest;
private ArrayList<LatLng> points;

Polyline line;

private static final float SMALLEST_DISPLACEMENT = 0.5F;
//half of a meter
```

6) The activity has just like any other, an onCreate method, we have to initialize our FusedLocationClient and create a location request. Add this to the code already there. Here we are doing typical onCreate type actions. Initialising the locationClient, obtaining an initial location, moving the map to that location, initialising an array of LatLng points to store our path as we move, setting up a callback for location updates.

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready
to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        mFusedLocationClient =
                LocationServices.getFusedLocationProviderClient(this);
// get the initial location
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        mFusedLocationClient.getLastLocation().addOnSuccessListener(this, new
                OnSuccessListener<Location>() {
                    @Override
                    public void onSuccess(Location location) {
                        LatLng loc = new LatLng(location.getLatitude(),
                                location.getLongitude());
                        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(loc,
19)); // zoom

                    }
                });

        // to hold our location readings
        points = new ArrayList<LatLng>();

        // on location updates
        locationCallback = new LocationCallback() {
            @Override
            public void onLocationResult(LocationResult locationResult) {
                if (locationResult == null) {
                    return;
                }
                for (Location location : locationResult.getLocations()) {
                    if (location != null) {
                        double latitude = location.getLatitude();
                        double longitude = location.getLongitude();
                        LatLng latLng = new LatLng(latitude, longitude);
                        points.add(latLng);
                        redrawLine();

                        mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
                    }
                }
            }
        };
        createLocationRequest(); // set up the location tracking
```

At this point you will notice that we need to have the google services built alongside our code. Go to the ‘Gradle Scripts’ section and insert into the app module dependencies

```
implementation 'com.google.android.gms:play-services-maps:17.0.0'
implementation 'com.google.android.gms:play-services-location:17.0.0'
```

Note: This is dependent on your system. Inserting this into our build scripts allows your app to have access to the resources provided by play-services (Maps and Location).

The last line of the onCreate method calls a method to create a location request.

Implement the createLocationRequest method.

```
protected void createLocationRequest() {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(2000);
    mLocationRequest.setFastestInterval(1000);
    mLocationRequest.setSmallestDisplacement(SMALLEST_DISPLACEMENT);

    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
        //                                          int[] grantResults)
        // to handle the case where the user grants the permission. See the
documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
```

Here, we set the request to be every 2 seconds, no shorter than 1seconds, of 0.5m minimum displacement and of high accuracy.

8) Ensure your onMapReady() methods looks like this, else you will get a marker to Sydney

```
9)@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode,
String[] permissions,
        //                                          int[] grantResults)
        // to handle the case where the user grants the permission. See the
documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
    if (!mFusedLocationClient.getLastLocation().isSuccessful()) {
        createLocationRequest(); // set up the location tracking
    }
    // set to current location

}
```

10) Go back to the code where we implemented the callback, note, we call a method redrawLine() to overlay the map with our path.

Continue to last page.

11) To get our lines to appear for each sampled location we implement the custom method redrawLine()

```
private void redrawLine(){
   mMap.clear(); //clears all overlays
   PolylineOptions options = new
          PolylineOptions().width(5).color(Color.BLUE).geodesic(true);
   for (int i = 0; i < points.size(); i++) {        LatLng point = points.get(i);
options.add(point);
   }
   line = mMap.addPolyline(options); //adds Polyline }
```

If you run the app and don’t see a map, it would be either, the API key is invalid, you have no network connectivity or you have not set the permissions for the app in the setting menu□Apps of your device.