



Lập trình Android

Bài 24: *Google Maps API*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-cSDL>

2019





Nội dung

1. Google Maps API
2. Thêm bản đồ Google Maps vào ứng dụng
3. Thêm Marker vào bản đồ
4. Các kiểu bản đồ
5. Làm việc với Geocoding
6. Vẽ hình lên bản đồ

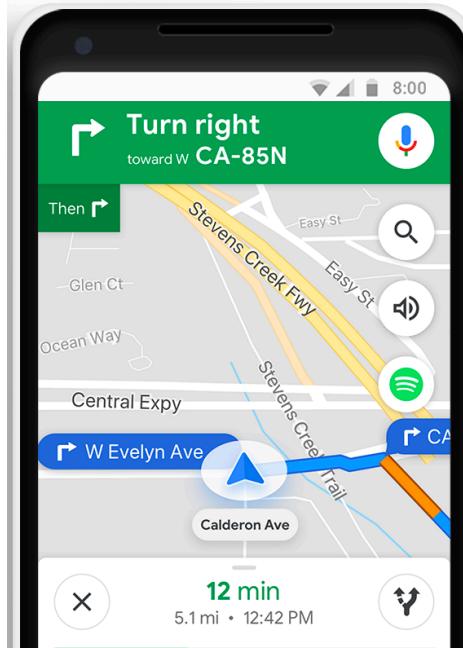




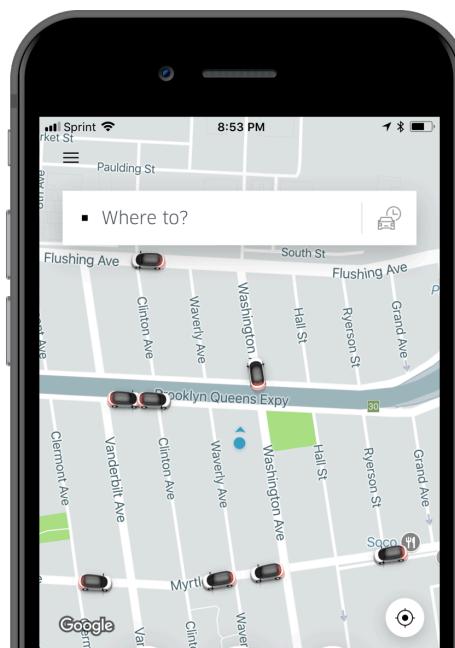
Google Maps API

- Có thể thêm dữ liệu bản đồ vào app nhờ **Google Maps API** cho Android
- API này tự động xử lý truy cập Google Maps server, tải dữ liệu, hiển thị bản đồ và tương tác UI với bản đồ
- API này còn cho phép thêm vào bản đồ các marker, polygon...
- Google Maps API được sử dụng trong nhiều ứng dụng khác nhau như: điều hướng (navigation), xác định vị trí (position tracking), các địa điểm lân cận (near by) ...

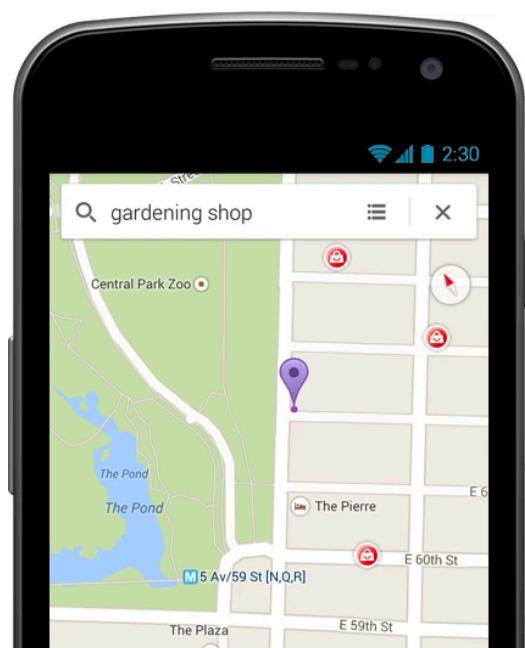
Navigation



Position Tracking



Near by





Thiết lập Google Maps API

- Để tích hợp Google Maps API vào project, truy cập vào URL sau:
<https://developers.google.com/maps/documentation/android-api/>
- Chọn **Get Started** và điền các thông tin theo yêu cầu; copy **API KEY** để đưa vào Android project

Enable Google Maps Platform

To enable APIs or set up billing, we'll guide you through a few tasks:

1. Pick product(s) below

2. Select a project

3. Set up your billing

Maps

Build customized map experiences that bring the real world to your users.

Routes

Give your users the best way to get from A to Z.

Places

Help users discover the world with rich details.

CANCEL

CONTINUE



Enable Google Maps Platform

You're all set!

You're ready to start developing!

YOUR API KEY

AIza:



To improve your app's security, restrict this key's usage in the [API Console](#).

DONE





Thiết lập Google Maps API (2)

- Tạo thuộc tính `google_maps_api_key` trong file `res/values/strings.xml` chứa API KEY
- Tạo thẻ metadata trong thẻ application của `AndroidManifest.xml` để khai báo sử dụng **Google Maps API** với key tương ứng

```
<application ...>

    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />

    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="@string/google_maps_api_key" />
</application>
```

- Cấu hình **Gradle build file**

```
dependencies {
    implementation 'com.google.android.gms:play-services:12.0.1'
}
```





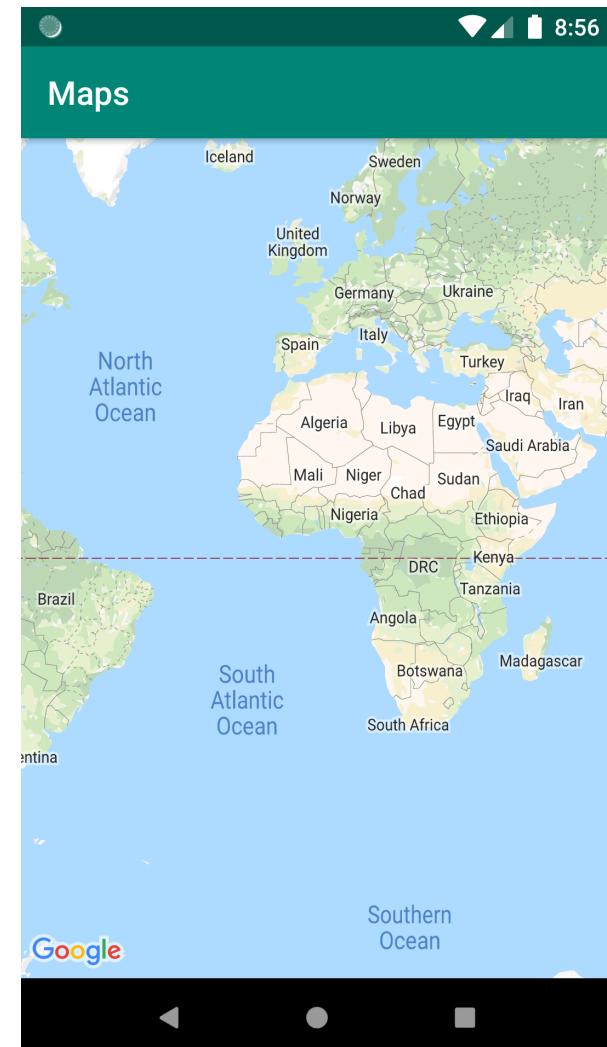
Thêm Map vào ứng dụng

- Thêm fragment mới vào layout activity, có name là **com.google.android.gms.maps.SupportMapFragment**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/frgMaps"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```





Thêm Map vào ứng dụng (2)

```
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback { 1  
    GoogleMap map;
```

```
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

(1) **MainActivity** implements **OnMapReadyCallback** và override phương thức **onMapReady**

```
2    SupportMapFragment mapFragment  
        = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.frgMaps);  
        mapFragment.getMapAsync(this);  
    }
```

(2) **SupportMapFragment** gọi hàm **getMapAsync** để đồng bộ dữ liệu bản đồ và xác định thời điểm bản đồ sẵn sàng. Khi bản đồ sẵn sàng, phương thức **onMapReady** sẽ được gọi

```
    @Override  
    public void onMapReady(GoogleMap googleMap) {  
        if (googleMap != null) {  
            map = googleMap;  
            // ...  
        }  
    }  
}
```



Thêm Marker vào bản đồ

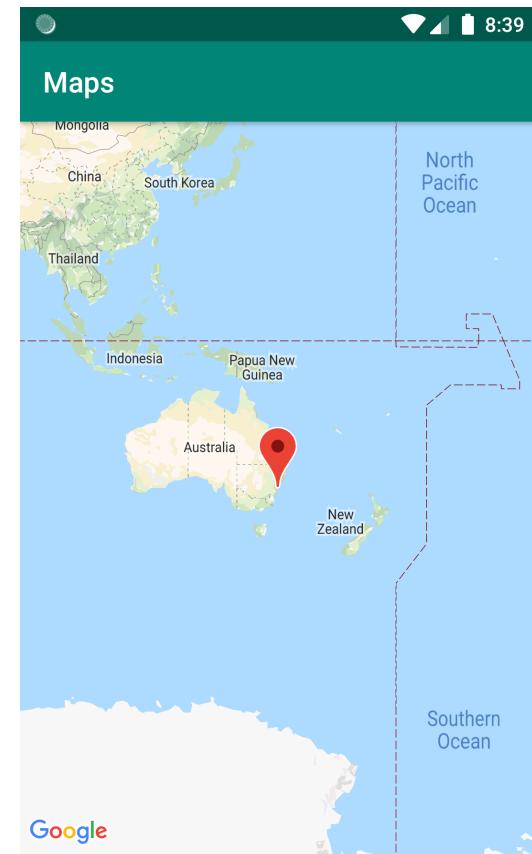
- Marker là đánh dấu một vị trí trên bản đồ, giúp người dùng tương tác với vị trí tương ứng
- Tùy đổi tượng **GoogleMap** có được từ **onMapReady**, có thể thêm vào một marker với các thông tin được đặc tả trong **MarkerOptions** như position, title... tại toạ độ **LatLng**

```
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {
```

```
    // ...
```

```
    @Override
```

```
    public void onMapReady(GoogleMap googleMap) {
        if (googleMap != null) {
            map = googleMap;
            LatLng sydney = new LatLng(-34.0, 151.0);
            map.addMarker(new MarkerOptions()
                .position(sydney).title("Marker in Sydney"));
            map.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        }
    }
}
```





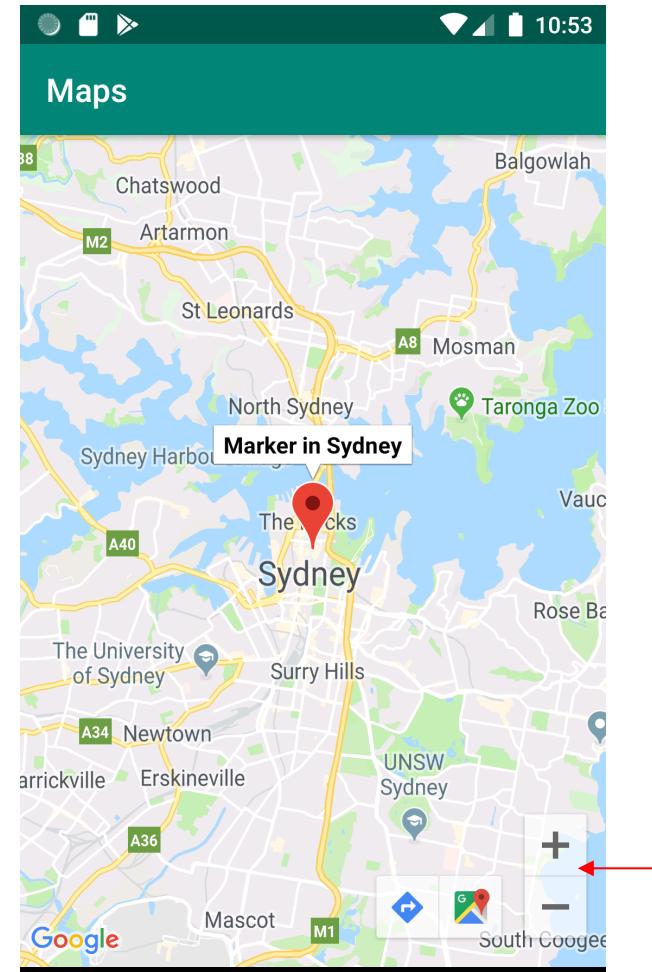
Thêm Marker vào bản đồ (2)

- **Zoom** vào vị trí đặt marker:

```
map.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 12));
```

- Hiển thị **zoom controls** (zoom in +, zoom out -):

```
map.getUiSettings().setZoomControlsEnabled(true);
```





Thêm Marker vào bản đồ (3)

- Xử lý **click event** lên marker view:

- Để MainActivity implement từ interface **GoogleMap.OnMarkerClickListener**:

```
public class MainActivity extends AppCompatActivity
    implements OnMapReadyCallback, GoogleMap.OnMarkerClickListener
```

- Override phương thức **onMarkerClick**:

```
@Override
public boolean onMarkerClick(Marker marker) {
    return false;
}
```

- Thiết lập **onMarkerClickListener** cho GoogleMap là MainActivity

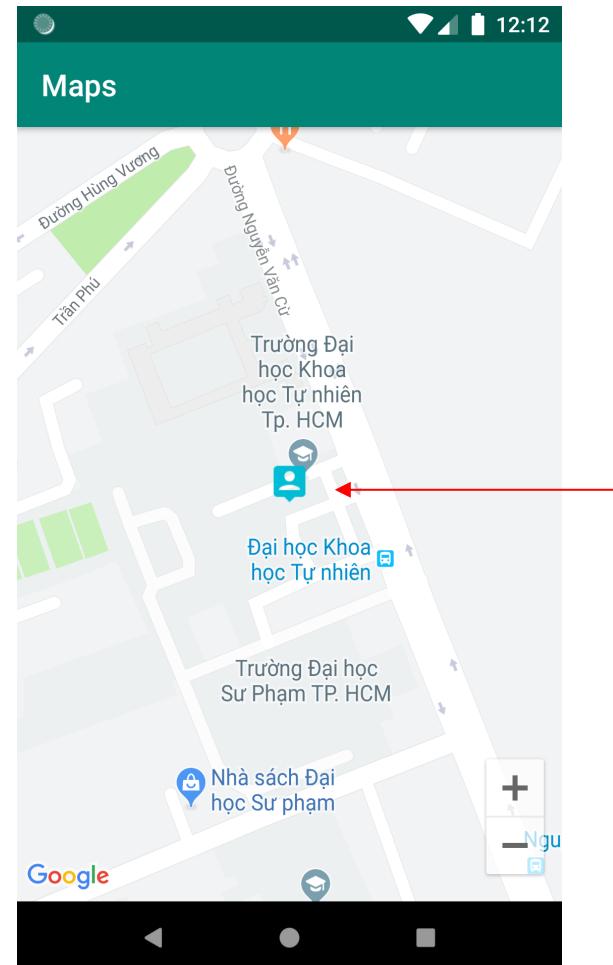
```
map.setOnMarkerClickListener(this);
```



Thêm marker vào bản đồ (4)

- Có thể thay đổi icon cho marker bằng cách tùy chỉnh **MarkerOptions**:

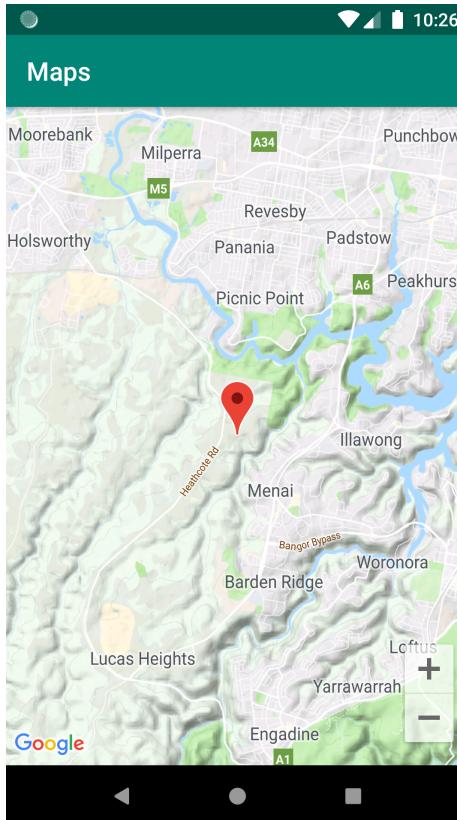
```
options.icon(BitmapDescriptorFactory.fromBitmap(  
    BitmapFactory.decodeResource(getResources(), R.drawable.ic_user_location)));
```



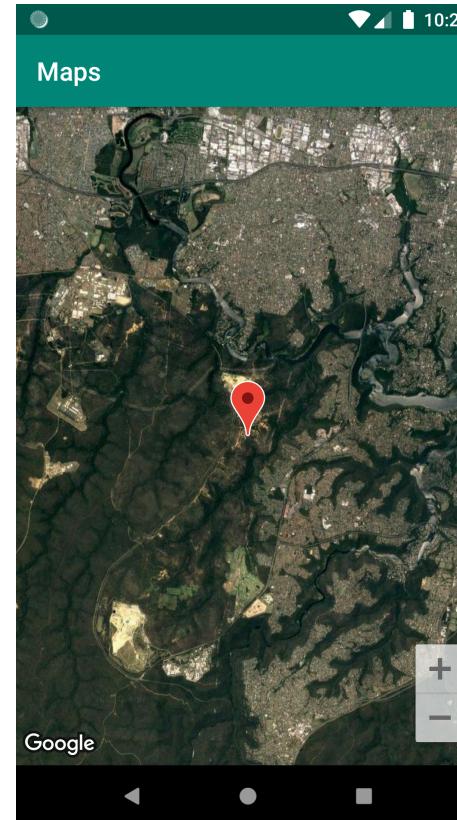


Các kiểu bản đồ

- Google Maps API cung cấp 4 kiểu hiển thị bản đồ:
MAP_TYPE_NORMAL, MAP_TYPE_SATELLITE, MAP_TYPE_TERRAIN, MAP_TYPE_HYBRID
- Thay đổi kiểu bản đồ bằng phương thức **setMapType** của đối tượng GoogleMap:
`map.setMapType(GoogleMap.MAP_TYPE_HYBRID);`

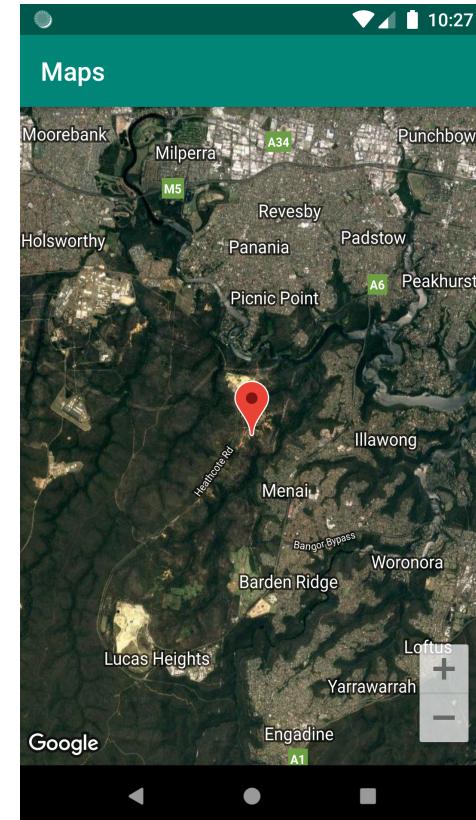


MAP_TYPE_TERRAIN



MAP_TYPE_SATELLITE

Lập trình Android



MAP_TYPE_HYBRID



Làm việc với Geocoding

- Google cung cấp class **Geocoder** để xác định địa chỉ của một vị trí và ngược lại

```
private String getAddress(LatLng latLng) {  
    // 1 – Tạo đối tượng Geocoder để xác định địa chỉ từ vị trí và ngược lại  
    Geocoder geocoder = new Geocoder(this);  
    List<Address> addresses;  
    Address address;  
    String addressText = "";  
    try {  
        // 2 – Hỏi Geocoder để lấy địa chỉ từ vị trí  
        addresses = geocoder.getFromLocation(latLng.latitude, latLng.longitude, 1);  
        // 3 – Nếu kết quả có chứa thông tin địa chỉ, thực hiện lấy chuỗi địa chỉ  
        if (null != addresses && !addresses.isEmpty()) {  
            address = addresses.get(0);  
            if (address.getMaxAddressLineIndex() > 0) {  
                for (int i = 0; i < address.getMaxAddressLineIndex(); i++) {  
                    addressText += (i == 0) ? address.getAddressLine(i) : "\n" + address.getAddressLine(i);  
                }  
            } else {  
                addressText = address.getAddressLine(0);  
            }  
        }  
    } catch (IOException e) {  
        Log.e("MainActivity", e.getLocalizedMessage());  
    }  
    return addressText;
```





Làm việc với Geocoding (2)

- Ví dụ, với khi đặt một **marker** của một vị trí lên bản đồ, có thể thiết lập title bằng địa chỉ của vị trí đó

```
private void placeMarkerOnMap(LatLng location) {  
    MarkerOptions options = new MarkerOptions().position(location);  
    String address = getAddress(location);  
    options.title(address);  
    map.addMarker(options);  
}
```





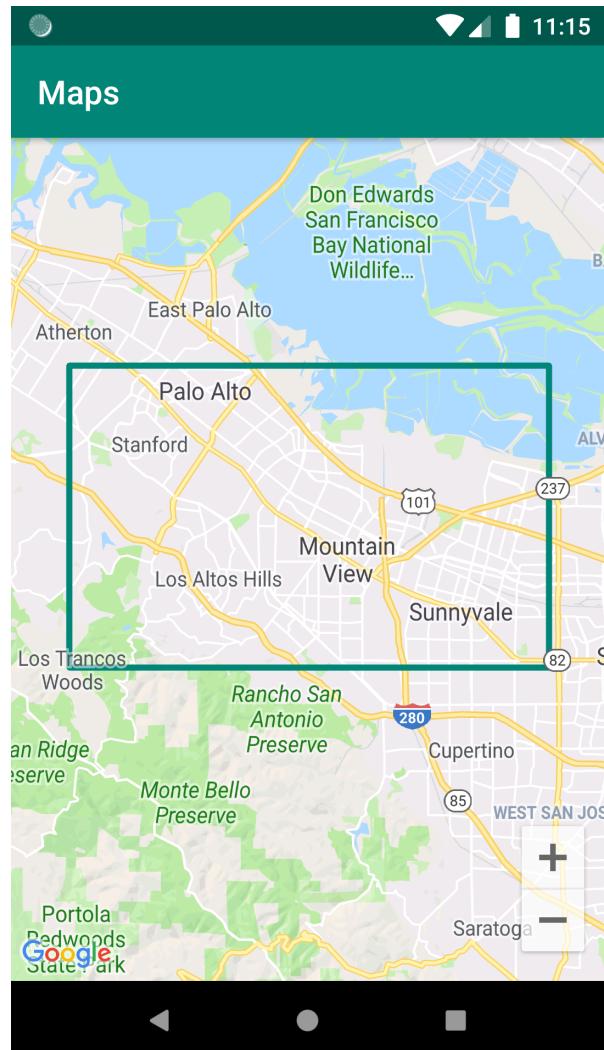
Vẽ hình lên bản đồ - Polylines

- Có thể nối các điểm thành một polyline bằng cách tạo đối tượng **PolylineOptions** và thêm các điểm cần nối vào

```
PolylineOptions rectOptions = new PolylineOptions()
```

```
    .add(new LatLng(37.35, -122.0))  
    .add(new LatLng(37.45, -122.0))  
    .add(new LatLng(37.45, -122.2))  
    .add(new LatLng(37.35, -122.2))  
    .add(new LatLng(37.35, -122.0))  
    .color(getResources().getColor(R.color.colorPrimary, null));
```

```
Polyline polyline = map.addPolyline(rectOptions);
```





Vẽ hình lên bản đồ - Polygons

- Để vẽ một đa giác lên bản đồ bằng cách xác định tạo đối tượng **PolygonOptions** và thêm các điểm cần nối tạo thành đa giác

```
PolygonOptions rectOptions = new PolygonOptions()
```

```
    .add(new LatLng(37.35, -122.0),  
        new LatLng(37.45, -122.0),  
        new LatLng(37.45, -122.2),  
        new LatLng(37.35, -122.2),  
        new LatLng(37.35, -122.0))  
    .strokeColor(Color.RED).fillColor(Color.BLUE);
```

```
Polygon polygon = map.addPolygon(rectOptions);
```





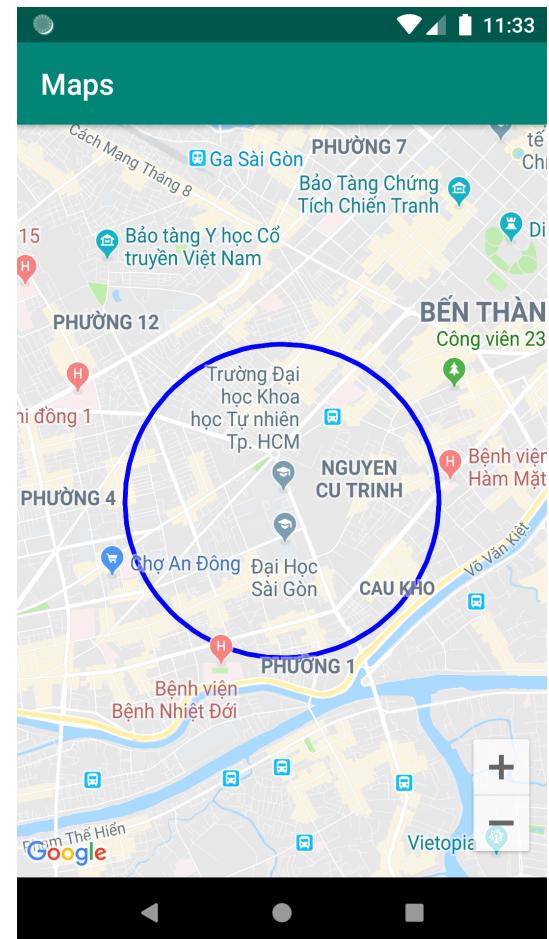
Vẽ hình lênh bản đồ - Circles

- Vẽ một đường tròn lên bản đồ theo toạ độ cho trước, bán kính theo đơn vị mét

```
CircleOptions circleOptions = new CircleOptions()
```

```
    .center(new LatLng(10.762707022680969, 106.68207389990175))  
    .radius(1000)  
    .strokeColor(Color.BLUE);
```

```
Circle circle = map.addCircle(circleOptions);
```





Q&A

