

技术学习报告

王思全 14331264 电子政务

- 百度地图 API

- 通过使用百度地图 SDK 可以在程序中内置百度地图

```
Bitmap bitmap = Bitmap.createScaledBitmap(BitmapFactory.decodeResource(getResources(),
    R.mipmap.pointer), 100, 100, true);
BitmapDescriptor bitmapD = BitmapDescriptorFactory
    .fromBitmap(bitmap);

mBMapView.getMap().setMyLocationEnabled(true);
MyLocationConfiguration config = new MyLocationConfiguration(
    MyLocationConfiguration.LocationMode.NORMAL, true, bitmapD);
mBMapView.getMap().setMyLocationConfiguration(config);

MyLocationData.Builder data = new MyLocationData.Builder();
data.latitude(desLatLng.latitude);
data.longitude(desLatLng.longitude);
data.direction(mSensorUtils.getCurrentRotation());

mBMapView.getMap().setMyLocationData(data.build());
```

- 通过对于指定坐标的设定，着色，可以实现轨迹展示
 - 每次随机生成坐标点，转化成百度地图标注点，加入到地图中，再和上一次的点连接成一条线。

```
public void addCustomElementsDemo() {
    // 添加折线
    LatLng p1 = new LatLng(39.97923, 116.357428);
    LatLng p2 = new LatLng(39.94923, 116.397428);
    /* LatLng p3 = new LatLng(39.97923, 116.437428);
    LatLng p4 = new LatLng(39.95923, 116.467428);
    LatLng p5 = new LatLng(39.96923, 116.487428);*/

    points = new ArrayList<LatLng>();
    points.add(p1);
    points.add(p2);
    /* points.add(p3);
    points.add(p4);
    points.add(p5);*/

    OverlayOptions ooPolyline = new PolylineOptions().width(12)
        .color(0xAAFF0000).points(points);

    mMarkerPolyLine = (Polyline) mBaiduMap.addOverlay(ooPolyline);

    OverlayOptions ooA = new MarkerOptions().position(p2).icon(bdA);
    mMarkerA = (Marker) (mBaiduMap.addOverlay(ooA));
}
```

- Navigation Bar 导航栏

- 主要利用了 drawerlayout

- 需要三个文件

- ◆ Activity.xml

```
<include layout="@layout/content_main" />

<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_main"
    app:menu="@menu/activity_main_drawer" />
```

- ◆

- ◆ Content.xml

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="16dp"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="16dp"
app:layout_behavior="android.support.design.widget.Ap
tools:context="com.example.tryvic.MainActivity"
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:id="@+id/tw" />
```

- ◆

- ◆ Drawer.xml

```

<group android:checkableBehavior="single">
    <item
        android:id="@+id/nav_camera"
        android:icon="@drawable/ic_menu_camera"
        android:title="Import" />
    <item
        android:id="@+id/nav_gallery"
        android:icon="@drawable/ic_menu_gallery"
        android:title="Gallery" />
    <item
        android:id="@+id/nav_slideshow"
        android:icon="@drawable/ic_menu_slideshow"
        android:title="Slideshow" />
    <item
        android:id="@+id/nav_manage"
        android:icon="@drawable/ic_menu_manage"
        android:title="Tools" />
</group>

<item android:title="Communicate">
    <menu>
        <item
            android:id="@+id/nav_share"
            android:icon="@drawable/ic_menu_share"
            android:title="Share" />
        <item
            android:id="@+id/nav_send"
            android:icon="@drawable/ic_menu_send"
            android:title="Send" />
    </menu>
</item>

```

- ◆
 - Drawer 负责导航栏的具体内容，content 负责页面信息，而 activity 将两者结合起来，展示给用户
- Broadcast 广播
 - 广播分成两类，一种是静态广播，一种是动态广播
 - 在这次的工程中，为了更加灵活的处理各种报警信息，我们决定使用动态广播的技术
 - 动态广播需要建立一个动态广播的 class

```

public class BrocastRec extends BroadcastReceiver {
    private static final String DYNAMIC = "com.example.tryvic.brocastrec";
    @Override
    public void onReceive(Context context, Intent intent) {
        if(intent.getAction().equals(DYNAMIC)) {
            Bundle bundle = intent.getExtras();
            String str = bundle.getString("message");

            NotificationManager nm = (NotificationManager) context.getSystemService(context.NOTIFICATION_SERVICE);

            Notification.Builder builder = new Notification.Builder(context);
            //Bitmap bm = BitmapFactory.decodeResource(context.getResources(), R.mipmap.dynamic);
            Intent notificationIntent = new Intent(context, MainActivity.class); // 点击该通知后要跳转的Activity
            PendingIntent contentIntent = PendingIntent.getActivity(context, 0, notificationIntent, 0);

            builder.setContentTitle("警报广播").setContentText(str).setTicker("警报WARNING").setSmallIcon(R.mipmap.ic_launcher)
                .setAutoCancel(true).setContentIntent(contentIntent);

            Notification notification = builder.build();

```

-
- 这主要是在 app 中动态注册了一个监听器

```

public void broadcast_f() {
    IntentFilter filter = new IntentFilter();
    filter.addAction("com.example.tryvic.brocastrec");
    registerReceiver(myReceiver, filter);
    Intent intent= new Intent("com.example.tryvic.brocastrec");
    String str = b_t.getText().toString();
    //str_all= str;
    Bundle bundle= new Bundle();
    bundle.putString("message", str);
    intent.putExtras(bundle);
    sendBroadcast(intent);
}

```

-
- 随后在代码中实现对监听器的使用，就实现了广播