

手势处理

1.1 知识解析

利用触摸屏的 Fling、Scroll 等 Gesture（手势）操作来操作会使得应用程序的用户体验大大提升，比如用 Scroll 手势在浏览器中滚屏，用 Fling 在阅读器中翻页等。

在 Android 中，手势的识别是通过 GestureDetector.OnGestureListener 接口来实现的。它有一些用于处理手势的方法需要被覆盖：

- `boolean onDown(MotionEvent e)`：用户轻触触摸屏，由 1 个 MotionEvent ACTION_DOWN 触发
- `void onShowPress(MotionEvent e)`：用户轻触触摸屏，尚未松开或拖动，由一个 1 个 MotionEvent ACTION_DOWN 触发
- `boolean onSingleTapUp(MotionEvent e)`：用户（轻触触摸屏后）松开，由一个 1 个 MotionEvent ACTION_UP 触发
- `boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)`：用户按下触摸屏、快速移动后松开，由 1 个 MotionEvent ACTION_DOWN，多个 ACTION_MOVE，1 个 ACTION_UP 触发
 - `e1`：第 1 个 ACTION_DOWN MotionEvent
 - `e2`：最后一个 ACTION_MOVE MotionEvent
 - `velocityX`：X 轴上的移动速度，像素/秒
 - `velocityY`：Y 轴上的移动速度，像素/秒
- `void onLongPress(MotionEvent e)`：用户长按触摸屏，由多个 MotionEvent ACTION_DOWN 触发
- `boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)`：用户按下触摸屏，并拖动，由 1 个 MotionEvent ACTION_DOWN，多

个 ACTION_MOVE 触发

使用 GestureDetector

1. 实现 OnGestureListener，实现相关的方法

2. 在需要使用手势识别的组件上调用 setOnTouchListener() 方法注册监听器

在 OnTouchListener 的 onTouch() 方法中，将触摸事件转交给 GestureDetector 的 onTouchEvent 方法来处理

3. 在组件上使用 setLongClickable(true)，使其能接受长按事件（必须要有）

也可以在对应组件的 XML 中使用 android:longClickable="true" 来实现

1.2 功能演示



1.3 实战操作

```
public class MainActivity extends Activity implements
OnGestureListener, OnTouchListener {
    ImageView iv;
    private GestureDetector mGestureDetector;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mGestureDetector = new GestureDetector(this, this);
        iv = (ImageView) findViewById(R.id.imageView1);
        iv.setOnTouchListener(this);
        // 要让其可以接受手势滑动, 必须让其能接受长按事件
        iv.setLongClickable(true);
    }

    @Override
    public boolean onDown(MotionEvent arg0) {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float
velocityX, float velocityY) {
        android.util.Log.e("", e1.getX() + "....." + e2.getX());
        // TODO Auto-generated method stub
        if (e1.getX() - e2.getX() > 20) {
            iv.setImageResource(R.drawable.lrzx);
        } else if (e1.getX() - e2.getX() < -20) {
            iv.setImageResource(R.drawable.nyn);
        }
        return false;
    }
}
```

```
}

@Override
public void onLongPress(MotionEvent e) {
    // TODO Auto-generated method stub

}

@Override
public boolean onScroll(MotionEvent e1, MotionEvent e2, float
distanceX, float distanceY) {
    // TODO Auto-generated method stub
    android.util.Log.e("", "Scroll...");

    return false;
}

@Override
public void onShowPress(MotionEvent e) {
    // TODO Auto-generated method stub

}

@Override
public boolean onSingleTapUp(MotionEvent e) {
    // TODO Auto-generated method stub
    return false;
}

// onTouchListener的方法
@Override
public boolean onTouch(View v, MotionEvent event) {
    // 转交给OnGestureListener的onTouchEvent()处理
    return mGestureDetector.onTouchEvent(event);
}
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return false;
}

}
```

1.4 职业素质

利用触摸屏的 Fling、Scroll 等 Gesture（手势）操作来操作会使得应用程序的用户体验大大提升，比如用 Scroll 手势在 浏览器中滚屏，用 Fling 在阅读中翻页等。