[Обработка жестов в Android](http://ru.stackoverflow.com/questions/466165/%d0%9e%d0%b1%d1%80%d0%b0%d0%b1%d0%be%d1%82%d0%ba%d0%b0-%d0%b6%d0%b5%d1%81%d1%82%d0%be%d0%b2-%d0%b2-android)

|  |
| --- |
| Обработка жестов - тема очень обширная, здесь я попробую отметить только основные моменты.  Что касается сложных геометрических жестов (например жест - квадрат и тп.), то подход здесь следующий:   1. Для начала необходимо создать специальный файл, который будет хранить жесты и связывать их с идентификаторами. Создается такой файл с помощью утилиты *Gestures Builder*. 2. В самом приложении необходимо разместить на разметке слой, который будет "принимать" жесты: 3. <android.gesture.GestureOverlayView 4. android:id="@+id/gestures" 5. android:layout\_width="fill\_parent" 6. android:layout\_height="fill\_parent">   </android.gesture.GestureOverlayView>   1. Регистрируем на этот слой слушатель OnGesturePerformedListener, который будет обрабатывать жесты и через колбэк onGesturePerformed возвращать идентификаторы распознаных жестов.   Подробнее смотрите, например, [эту статью](http://habrahabr.ru/post/120016/).  Для обнаружения простых жестов, вроде смахивания, все гораздо проще.  Вам нужно повесить слушатель OnTouchListener на свой элемент, который должен "понимать" смахивания. В колбэке интерфейса onTouch() регистрируется GestureDetector.OnGestureListener, который позволяет переопределить метод onFling() - обнаружение смахивания - в этом методе вы определяете логику действий при смахивании. Так же интерфейс распознает еще [несколько действий](http://developer.android.com/intl/ru/reference/android/view/GestureDetector.OnGestureListener.html): скроллинг, долгое касание и некоторые другие  Простой [пример](http://stackoverflow.com/a/4098447):  public class MyActivity extends Activity {  private void onCreate() {  final GestureDetector gdt = new GestureDetector(new GestureListener());  final ImageView imageView = (ImageView) findViewById(R.id.image\_view);  imageView.setOnTouchListener(new OnTouchListener() {  @Override  public boolean onTouch(final View view, final MotionEvent event) {  gdt.onTouchEvent(event);  return true;  }  });  }  private static final int SWIPE\_MIN\_DISTANCE = 120;  private static final int SWIPE\_THRESHOLD\_VELOCITY = 200;  private class GestureListener extends SimpleOnGestureListener {  @Override  public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {  if(e1.getX() - e2.getX() > SWIPE\_MIN\_DISTANCE && Math.abs(velocityX) > SWIPE\_THRESHOLD\_VELOCITY) {  return false; // справа налево  } else if (e2.getX() - e1.getX() > SWIPE\_MIN\_DISTANCE && Math.abs(velocityX) > SWIPE\_THRESHOLD\_VELOCITY) {  return false; // слева направо  }  if(e1.getY() - e2.getY() > SWIPE\_MIN\_DISTANCE && Math.abs(velocityY) > SWIPE\_THRESHOLD\_VELOCITY) {  return false; // снизу вверх  } else if (e2.getY() - e1.getY() > SWIPE\_MIN\_DISTANCE && Math.abs(velocityY) > SWIPE\_THRESHOLD\_VELOCITY) {  return false; // сверху вниз  }  return false;  }  }  }  Более глубокое погружение в тему распознавания простых жестов - смотрите, например, [эту статью](http://www.ibm.com/developerworks/ru/library/j-mobileforthemasses2/index.html). |