Рисуем и распознаем жесты с помощью GestureBuilder

|  |
| --- |
| [Главная](http://learn-android.ru/index.html) » Рисуем и распознаем жесты с помощью GestureBuilder |
| В этом уроке мы поработаем с жестами. Создадим такое приложение, в котором пользователь может своим пальцем нарисовать некоторую каракулю и программа выдаст ему показатель сходства того, что он нарисовал с заданными в ней эталонными объектами. Мы зададим в базу (сделаем библиотеку) из 5 цифр: 1, 2, 3, 4, 5, применив им соответствующие названия и при рисовании пользователем цифры, например 2, программа выдаст, что с такой то точностью нарисована цифра 2, то есть наше приложение будет распознавать введенный жест.  Чтобы начать создавать эту программу, сначала нужно подготовить файл с эталонными жестами, с которыми будет происходить сравнение нарисованных пользователем. Для этого воспользуемся стандартным Android приложением по имени **Gesture Builder**, которое появилось в Android OS с версии 1.6:  Созданные жесты цифр  Суть приложения заключается в том, что вы можете создавать там любые жесты и присваивать им названия, файл с созданными жестами сохраняется на SD карту устройства под названием **gestures**. Нам нужно создать такой файл, извлечь его с SD карты устройства на компьютер и положить его в файлы нашего проекта в специально созданную папку **res/raw/gestures**:  Папка для файла gestures  Я нарисовал 5 цифр от 1 до 5 и дал им соответствующие имена. Чтобы облегчить вам судьбу, выкладываю файл[gestures](http://learn-android.ru/Files/gestures.noext) моих 5 цифр. Но вы можете создать и свой файл, установив на устройство программу Gesture Builder.  Ну а теперь за дело. [Создаем](http://learn-android.ru/news/rabochaja_sreda_android_studio_i_hellow_development_world/2015-01-03-5.html) новый проект, выбираем Blank Activity, создаем папку raw и закидываем туда файл gestures (как на скриншоте). Интерфейс программы будет состоять из области для рисования жестов, задаваемого с помощью **android.gesture.GestureOverlayView** и текстового поля TextView, в котором отобразятся результаты сравнения нарисованных жестов с эталонными в файле gestures. Открываем файл **activity\_main.xml** и добавляем эти элементы:  <?xml version="1.0" encoding="utf-8"?>  <**LinearLayout** xmlns:android="schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <**TextView**  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:textSize="12pt"  android:text="Нарисуйте цифру от 1 до 5"  android:layout\_margin="5dip"/>  <**android.gesture.GestureOverlayView**  android:id="@+id/draw\_gestures"  android:layout\_width="fill\_parent"  android:layout\_height="0dip"  android:layout\_weight="1.0" />  <**TextView**  android:id="@+id/results"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:textSize="12pt"  android:layout\_margin="5dip"/>  </**LinearLayout**>  Теперь переходим к написанию кода приложения. Мы создали объект GestureOverlayView и ему нужно задать **OnGesturePerformedListener**, который будет отслеживать рисование каракулей пользователя. Когда рисование жеста выполнится, этот жест передается в метод **onGesturePerformed ()**, в котором и происходит сравнение введенного жеста со всеми формами, имеющимися в библиотеке gestures, а потом возвращает ответ в виде списка, в котором указывается коэффициент схожести нарисованного пользователем с каждым имеющимся в библиотеке. Каждый ответ сравнения из этого списка содержит имя сравниваемого жеста, задаваемый при его создании в приложении Gesture Builder и оценку его схожести с тем, что было нарисовано. Как правило, если эта оценка больше 1, то есть хорошее сходство.  Итак, открываем файл **MainActivity.java** и вводим необходимый код:  import **java.text.DecimalFormat**;  import **java.text.NumberFormat**;  import **java.util.ArrayList**;  import **android.app.Activity**;  import **android.gesture.Gesture**;  import **android.gesture.GestureLibraries**;  import **android.gesture.GestureLibrary**;  import **android.gesture.GestureOverlayView**;  import **android.gesture.Prediction**;  import **android.gesture.GestureOverlayView.OnGesturePerformedListener**;  import **android.os.Bundle**;  import **android.widget.TextView**;  **public** **class** MainActivity **extends** Activity **implements** OnGesturePerformedListener {  **private** **TextView** mResults;  **private** **GestureLibrary** mGestureLibrary;  **@Override**  **public** **void** onCreate(**Bundle** *savedInstanceState*) {  super.onCreate(savedInstanceState);  setContentView(**R**.layout.activity\_main);  *//Связываемся с нашими объектами интерфейса:*  mResults=(**TextView**)findViewById(**R**.id.results);  *//Инициализируем библиотеку жестов, которые мы создали*  *//в программе GestureBuilder и взяли оттуда файл с жестами:*  mGestureLibrary=**GestureLibraries**.fromRawResource(this, **R**.raw.gestures);  if (!mGestureLibrary.load()) finish();  *//Инициализируем объект GestureOverlayView интерфейса приложения*  *//и устанавливаем для него слушателя нарисованных жестов:*  **GestureOverlayView** gestures=(**GestureOverlayView**)findViewById(**R**.id.draw\_gestures);  gestures.addOnGesturePerformedListener(this);  }  *//Обрабатываем рисование жестов в области GestureOverlayView:*  **public** **void** onGesturePerformed(**GestureOverlayView** *overlay*,**Gesture** *gesture*) {  *//Здесь выполняется сравнение нарисованных пользователем*  *//каракулей с заданными в библиотеке жестов (файле gestures), и после сравнения*  *//выдается массив с цифрами, указывающими наибольшее сходство нарисованного с заданным в файле:*  **ArrayList** <**Prediction**> predictions = mGestureLibrary.recognize(gesture);  **String** predictList = "";  **NumberFormat** formatter = new **DecimalFormat**("#0.00");  for(**int** i=0; i<predictions.size(); i++) {  **Prediction** prediction = predictions.get(i);  *//Отображается имя объектов с файла gestures и оценка их*  *//сходства с нарисованным пользователем знаком в цифрах:*  predictList = predictList + prediction.name + " "  + formatter.format(prediction.score) + "\n";  }  *//Результаты сравнений отображаем в элементе TextView:*  mResults.setText(predictList);  }  }  Вот такая реализация приложения по рисованию и распознаванию жестов в Android. Теперь запускаем приложение и смотрим, что же получилось. Я ввел кракозябру, похожую на цифру 1, и получил вот такой ответ:  Вот что получилось в результате сравнения  Надеюсь, что вам это понравилось и что оно пригодится! |