**Android обработка жестов: как добавить жесты в свое приложение и распознать их**

Современные телефоны предоставляют программистам поистине удивительные возможность. Судите сами, сегодня дисплей для смартфона - не только средство отображения информации, но и мощный интерфейс для взаимодействия с пользователем. Современные дисплеи могут фиксировать множество прикосновений, распознавать жесты и даже определять силу нажатия (3D Touch). Тем удивительнее, что все это богатство довольно редко используется программистами при разработке новых игр и мобильных приложений. Эта статья поможет исправить это досадное недоразумение. Сегодня речь пойдет о распознавании жестов.

В Android для поддержки жестов используется специальный вид: "GestureOverlayView". Жесты как таковые представляют собой двоичные ресурсы. Их можно создать с помощью Android SDK. В свою activity жесты можно загрузить с помощью метода GestureLib.fromRawResource(). Если система распознает жест, вызывается метод onGesturePerformedListener(). Соответственно у activity должен быть объявлена поддержка интерфейса GestureOverlayView, кроме того, Activity должна зарегистрировать себя в GestureOverlayView с помощью метода addOnGesturePerformedListener().

Распознанные жестов в Android отображаются желтым цветом - а нераспознанные - бледно-желтым. Эту подсветку можно отключить через методы setGestureColor(Color.TRANSPARENT) и setUncertainGestureColor(Color.TRANSPARENT) для GestureOverlayView.

Создать новые жесты можно через Android эмулятор, запустив на нем программу GestureBuilder. Вы можете создать несколько жестов с одинаковыми именами. Это поможет более точно распознавать их. При создании эмулятора добавьте поддержку sdcard, в противном случае вы просто не сможете сохранить жесты. Все жесты сохраняются в файл gestures на эмуляторе. Скачать этот файл с эмулятора можно с помощью adb, выполнив команду:

./adb pull /sdcard/gestures ~/test

После ее выполнения файл с жестами будет сохранен в папку "res/raw" и их можно будет использовать в GestureOverlayView.

Если Вам не хочется возиться с эмулятором, можете установить на свой телефон программу ["Gesture Builder"](https://play.google.com/store/apps/details?id=pack.GestureApp) из Google Play. У нее очень простой интерфейс. В режиме Add вы водите пальцем по дисплею смартфона, рисуя нужный жест. После этого вы вводите его название, и жест сохраняется на карточку. При сохранении жеста отображается путь до файла gesture.txt. После добавления всех необходимых жестов этот файл нужно скопировать в свой проект.

Давайте разработаем небольшое тестовое приложение, реализующее распознавание жестов. Создайте новый проект. Главную Activity назовите "GestureTest". Создадим layout **main.xml**.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http:*//schemas.android.com/apk/res/android"*

    android:orientation="vertical"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    >

<TextView

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:text="@string/hello"/>

</LinearLayout>

Код главной activity приведен ниже. Обратите внимание, мы добавляем GestureOverlayView программно в методе onCreate. Там же мы привязали GesturePerformedListener для обработки жестов. Как видите, в onGesturePerformed мы создаем список predictions, содержащий наши жесты и параметр scope, который отражает степень соответствия введенного жеста жнсту библиотеки. В цикле for мы проходимся по этому списку, и если оказывается, что scope больше единицы, считаем жест распознанным, и через Toast печатаем сообщение на экране.

import java.util.ArrayList;

import android.app.Activity;

import android.gesture.Prediction;

import android.os.Bundle;

import android.view.View;

import android.widget.Toast;

import android.gesture.Gesture;

import android.gesture.GestureLibraries;

import android.gesture.GestureLibrary;

import android.gesture.GestureOverlayView;

import android.gesture.GestureOverlayView.OnGesturePerformedListener;

public class GestureTest extends Activity implements OnGesturePerformedListener {

  private GestureLibrary gestureLib;

  @Override

  public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    GestureOverlayView gestureOverlayView = new GestureOverlayView(this);

    View inflate = getLayoutInflater().inflate(R.layout.main, null);

    gestureOverlayView.addView(inflate);

    gestureOverlayView.addOnGesturePerformedListener(this);

    gestureLib = GestureLibraries.fromRawResource(this, R.raw.gestures);

    if (!gestureLib.load()) {

      finish();

    }

    setContentView(gestureOverlayView);

  }

  @Override

  public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture) {

    ArrayList<Prediction> predictions = gestureLib.recognize(gesture);

    for (Prediction prediction : predictions) {

      if (prediction.score > 1.0) {

        Toast.makeText(this, prediction.name, Toast.LENGTH\_SHORT)

            .show();

      }

    }

  }

}

Можете запустить приложение и поводить пальцем по дисплею своего смартфона, поиграться с различными жестами.