ЛЕКЦИЯ. ПРИЛОЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ (РЕЧИ, ЖЕСТОВ)

Начнем с распознавания речи.

Структура файла манифеста:

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

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>"

      package="com.jameselsey"

      android:versionCode="1"

      android:versionName="1.0">

    <application android:label=

"VoiceRecognitionDemo" android:icon="@drawable/icon"

            android:debuggable="true">

        <activity android:name=".VoiceRecognitionDemo"

                  android:label="@string/app\_name">

            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />

            </intent-filter>

        </activity>

    </application>

</manifest>

В папке layout создадим два .xml-файла res/layout/voice\_recog.xml :

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

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

    android:layout\_width="fill\_parent"

    android:layout\_height="fill\_parent"

    android:orientation="vertical">

    <TextView

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:paddingBottom="4dip"

        android:text="Click the button and start speaking" />

    <Button android:id="@+id/speakButton"

        android:layout\_width="fill\_parent"

        android:onClick="speakButtonClicked"

        android:layout\_height="wrap\_content"

        android:text="Click Me!" />

    <ListView android:id="@+id/list"

        android:layout\_width="fill\_parent"

        android:layout\_height="0dip"

        android:layout\_weight="1" />

 </LinearLayout>

Следует нажать кнопку Click Me! И начать говорить. Распознаваемые слова будут выводиться в списке ListView.

А также res/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="fill\_parent"

    android:layout\_height="fill\_parent"

    >

<TextView

    android:layout\_width="fill\_parent"

    android:layout\_height="wrap\_content"

    android:text="VoiceRecognition Demo!"

    />

</LinearLayout>

Главная активность:

package com.jameselsey;

import android.app.Activity;

import android.os.Bundle;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.content.pm.ResolveInfo;

import android.speech.RecognizerIntent;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import java.util.ArrayList;

import java.util.List;

/\*\*

 \* A very simple application to handle Voice Recognition intents

 \* and display the results

 \*/

public class VoiceRecognitionDemo extends Activity

{

    private static final int REQUEST\_CODE = 1234;

    private ListView wordsList;

    /\*\*

     \* Called with the activity is first created.

     \*/

    @Override

    public void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.voice\_recog);

        Button speakButton = (Button) findViewById(R.id.speakButton);

        wordsList = (ListView) findViewById(R.id.list);

        // Disable button if no recognition service is present

        PackageManager pm = getPackageManager();

        List<ResolveInfo> activities = pm.queryIntentActivities(

                new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH), 0);

        if (activities.size() == 0)

        {

            speakButton.setEnabled(false);

            speakButton.setText("Recognizer not present");

        }

    }

    /\*\*

     \* Handle the action of the button being clicked

     \*/

    public void speakButtonClicked(View v)

    {

        startVoiceRecognitionActivity();

    }

    /\*\*

     \* Fire an intent to start the voice recognition activity.

     \*/

    private void startVoiceRecognitionActivity()

    {

        Intent intent = new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);

        intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL,

                RecognizerIntent.LANGUAGE\_MODEL\_FREE\_FORM);

        intent.putExtra(RecognizerIntent.EXTRA\_PROMPT,

"Voice recognition Demo...");

        startActivityForResult(intent, REQUEST\_CODE);

    }

    /\*\*

     \* Handle the results from the voice recognition activity.

     \*/

    @Override

    protected void onActivityResult(int requestCode, int resultCode, Intent data)

    {

        if (requestCode == REQUEST\_CODE && resultCode == RESULT\_OK)

        {

            // Populate the wordsList with the String values the recognition engine thought it heard

            ArrayList<String> matches = data.getStringArrayListExtra(

                    RecognizerIntent.EXTRA\_RESULTS);

            wordsList.setAdapter(new ArrayAdapter<String>(this,

android.R.layout.simple\_list\_item\_1,

                    matches));

        }

        super.onActivityResult(requestCode, resultCode, data);

    }

}

packageManager проверяет наличие пакета, который выполняет трансляцию и обрабатывает intents для ACTION\_RECOGNIZE\_SPEECH.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Замечания**

Once you’ve checked whether the module is supported on your phone, carry on with the below steps to implement it successfully.

The heart of Speech to text in Android API is package android.speech and specially the class android.speech.RecognizerIntent.

|  |  |
| --- | --- |
| 3  4 | Intent intent = newIntent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);  intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL, "en-US"); |

Voice recognition feature can be achieved by RecognizerIntent. Create an Intent of type RecognizerIntent and pass the extra parameters and start activity for the result. It basically starts the recognizer prompt customized by your extra parameters. Internally, voice recognition communicates with the server and gets the results. So you must provide the internet access permission for the application.

Below described are the functions which are called above:

Call the RecognizerIntent – Your application calls the Voice Recognition Intent that records the voice.

ACTION\_RECOGNIZE\_SPEECH – Starts an activity that will prompt the user for speech and send it through a speech recognizer.

EXTRA\_LANGUAGE\_MODEL - Informs the recognizer that which speech model should be preferred when performing ACTION\_RECOGNIZE\_SPEECH. It is adapted to input messages in English.

**Permission Manifest File**

uses-permission android:name=”android.permission.INTERNET”

ЛАБОРАТОРНАЯ РАБОТА. РАСПОЗНАВАНИЕ ФИГУР (Gestures)

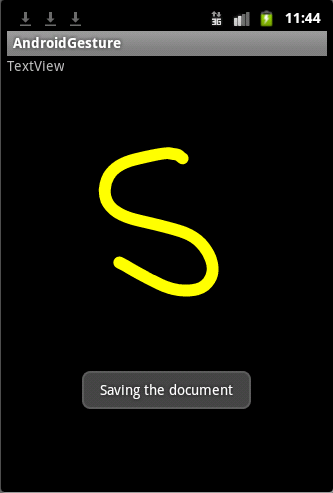
РАСПОЗНАВАНИЕ ФИГУР

Цель: рассмотреть средства Android для распознавания фигур.

Краткое теоретическое введение.

Прежде всего, нужно создать один или два файла с жестами. Для этой цели используем приложение Gestures Builder. Если оно не установлено, то его следует скачать с Google Play (приложение бесплатное). В программе понадобится специальное API – android.gesture.

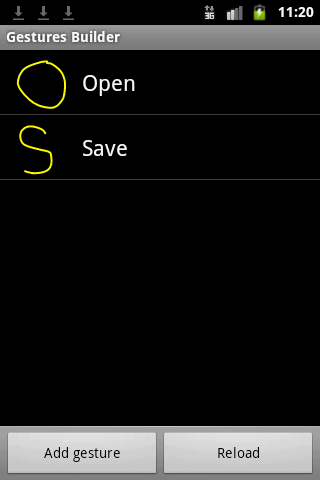
Итак, запустим Gesture Builder и создадим два жеста (буквы): “S” и “O”. При этом автоматически фигуры помещаются на SD карту в папку Gestures. Нам надо скопировать их из этой папки в проект – в папку res/raw (последнюю, возможно, понадобится создать). Когда соответствующий жест будет распознан, получим экран:

[](https://androidresearch.files.wordpress.com/2012/01/android-result.png)

**Создаем проект**: AndroidGesture  
**Activity**: AndroidGestureActivity

**1. Создаем жесты**  
Начиная с Андроид 1.6 Android Emulator includes содержит предустановленное приложение Gestures Builder.

Запустите Android Emulator и используйте Gesture Builder чтобы создать жесты “S” и “O”:

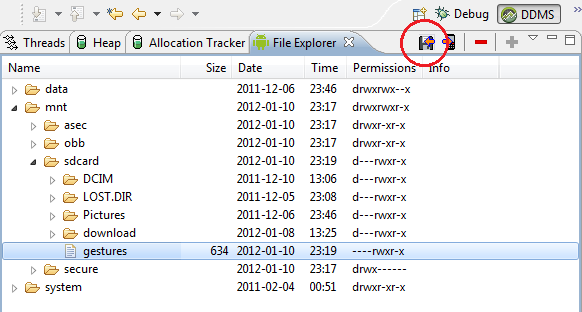
[](https://androidresearch.files.wordpress.com/2012/01/android-gestures.png)

Жест всегда ассоциируется с именем.

**2. Импорт жеста в проект**  
Всякий раз, когда мы создаем и редактируем жест с помощью Gesture Builder, создается файл на SD карте: /sdcard/gestures.Этот файл нужно импортировать в папку /res/raw.

Для этого откройте  **FileExplorer** в **DDMS** перспективе. (При необходимости добавьте FileExplorer tab из: Window -> Show View -> File Explorer). Зайдите а  /sdcard директорий и скопируйте gesture file в компьютер.

Копирование:

[](https://androidresearch.files.wordpress.com/2012/01/ddms.png)

Предварительно создайте папку ***raw*** в директории **res**

**3. Загрузите файл с жестом в эту папку gesture:**

Содержимое файла **main.xml**:

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

<android.gesture.GestureOverlayView xmlns:android="<http://schemas.android.com/apk/res/android>"

android:id="@+id/gestures"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:eventsInterceptionEnabled="true"

android:gestureStrokeType="multiple"

android:orientation="vertical" >

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="TextView" />

</android.gesture.GestureOverlayView>

Активити:

public class AndroidGestureActivity extends Activity

implements OnGesturePerformedListener {

GestureLibrary mLibrary;

@Override

public void onCreate(Bundle savedInstanceState) {

   super.onCreate(savedInstanceState);

   setContentView(R.layout.main);

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

   if (!mLibrary.load()) {

     finish();

   }

   GestureOverlayView gestures = (GestureOverlayView)

findViewById(R.id.gestures);

   gestures.addOnGesturePerformedListener(this);

}

public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture) {

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

   if (predictions.size() > 0 && predictions.get(0).score > 1.0) {

     String result = predictions.get(0).name;

     if ("open".equalsIgnoreCase(result)) {

       Toast.makeText(this, "Opening the document", Toast.LENGTH\_LONG).show();

     } else if ("save".equalsIgnoreCase(result)) {

       Toast.makeText(this, "Saving the document", Toast.LENGTH\_LONG).show();

     }

   }

}

Запустите приложение и проверьте его в работе. Система должна распознать рукописные символы и поместить распознанные значения в список:

ArrayList<Prediction> predictions

**ЗАДАНИЕ**.

1. Используя список predictions, проверьте наличие в нем какого-то определенного распознанного символа.
2. Подготовьте несколько различных вариантов записи одного и того же символа. Проверьте, как программа справляется с их распознаванием.
3. Постройте символ, который система не распознает.