

[RxJava \(http://developer.alexanderklimov.ru/android/rx/\)](http://developer.alexanderklimov.ru/android/rx/)

[Советы \(http://developer.alexanderklimov.ru/android/tips-android.php\)](http://developer.alexanderklimov.ru/android/tips-android.php)

[Статьи \(http://developer.alexanderklimov.ru/android/articles-android.php\)](http://developer.alexanderklimov.ru/android/articles-android.php)

[Книги \(http://developer.alexanderklimov.ru/android/books.php\)](http://developer.alexanderklimov.ru/android/books.php)

[Java \(http://developer.alexanderklimov.ru/android/java/java.php\)](http://developer.alexanderklimov.ru/android/java/java.php)

[Kotlin \(http://developer.alexanderklimov.ru/android/kotlin/\)](http://developer.alexanderklimov.ru/android/kotlin/)

[Дизайн \(http://developer.alexanderklimov.ru/android/design/\)](http://developer.alexanderklimov.ru/android/design/)

[Отладка \(http://developer.alexanderklimov.ru/android/debug/\)](http://developer.alexanderklimov.ru/android/debug/)

[Open Source \(http://developer.alexanderklimov.ru/android/opensource.php\)](http://developer.alexanderklimov.ru/android/opensource.php)

[Полезные ресурсы \(http://developer.alexanderklimov.ru/android/links.php\)](http://developer.alexanderklimov.ru/android/links.php)

Кто сказал Мяу? - работаем со звуками Му, Мяу, Гав

Напишем программу, которая поможет определить, кто-же сказал Мяу? Меня всегда интересовал данный вопрос.

Подготовим заранее картинки различных животных и вставим их в папку **res/drawable-xhdpi**. Создадим разметку с кнопками:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/gridLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:columnCount="2"
    android:rowCount="3"
    tools:context=".MainActivity">

    <ImageButton
        android:id="@+id/imageButtonCow"
        android:background="@android:color/transparent"
        android:src="@drawable/cow" />

    <ImageButton
        android:id="@+id/imageButtonChicken"
        android:background="@android:color/transparent"
        android:src="@drawable/chicken" />

    <ImageButton
        android:id="@+id/imageButtonCat"
        android:background="@android:color/transparent"
        android:src="@drawable/cat" />

    <ImageButton
        android:id="@+id/imageButtonDuck"
        android:background="@android:color/transparent"
        android:src="@drawable/duck" />

    <ImageButton
        android:id="@+id/imageButtonSheep"
        android:background="@android:color/transparent"
        android:src="@drawable/sheep" />

    <ImageButton
        android:id="@+id/imageButtonDog"
        android:background="@android:color/transparent"
        android:src="@drawable/dog" />

</GridLayout>

```

Разметка представляет собой сетку, в которой можно задать число колонок и рядов.

Положим подготовленные аудио-файлы с голосами животных в директорию **assets**. По умолчанию в проекте такой папки нет. Выбираем **File | New | Folder | Assets Folder**. В диалоговом окне оставляем всё без изменений и нажимаем кнопку **Finish**. Файлы, лежащие в этой папке, считайте тоже ресурсами. Но они имеют свои особенности, в частности вы можете создавать свою структуру подпапок.

Переходим к программной части. Нам надо создать объект **SoundPool**, загрузить в него аудио-файлы из папки **assets** методом **load()**.

Зададим максимальное количество одновременно проигрываемых потоков - 3.

При нажатии на кнопку будем проигрывать нужный звук.

```

package ru.alexanderklimov.saymeow;

import android.annotation.TargetApi;
import android.content.res.AssetFileDescriptor;
import android.content.res.AssetManager;
import android.media.AudioAttributes;
import android.media.AudioManager;
import android.media.SoundPool;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.MotionEvent;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;

import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private SoundPool mSoundPool;
    private AssetManager mAssetManager;
    private int mCatSound, mChickenSound, mCowSound, mDogSound, mDuckSound, mSheepSound;
    private int mStreamID;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //      if (android.os.Build.VERSION.SDK_INT < Build.VERSION_CODES.LOLLIPOP) {
        //          // Для устройств до Android 5
        //          createOldSoundPool();
        //      } else {
        //          // Для новых устройств
        //          createNewSoundPool();
        //      }

        mAssetManager = getAssets();
        //
        //      // получим идентификаторы
        //      mCatSound = loadSound("cat.ogg");
        //      mChickenSound = loadSound("chicken.ogg");
        //      mCowSound = loadSound("cow.ogg");
        //      mDogSound = loadSound("dog.ogg");
        //      mDuckSound = loadSound("duck.ogg");
        //      mSheepSound = loadSound("sheep.ogg");

        ImageButton cowImageButton = (ImageButton) findViewById(R.id.imageButtonCow);
        //      cowImageButton.setOnClickListener(onClickListener);

        ImageButton chickenImageButton = (ImageButton) findViewById(R.id.imageButtonChicken);
        chickenImageButton.setOnClickListener(onClickListener);

        ImageButton catImageButton = (ImageButton) findViewById(R.id.imageButtonCat);
        catImageButton.setOnClickListener(onClickListener);
    }
}

```

```

ImageButton duckImageButton = (ImageButton) findViewById(R.id.imageButtonDuck);
duckImageButton.setOnClickListener(onClickListener);

ImageButton sheepImageButton = (ImageButton) findViewById(R.id.imageButtonSheep);
sheepImageButton.setOnClickListener(onClickListener);

ImageButton dogImageButton = (ImageButton) findViewById(R.id.imageButtonDog);
dogImageButton.setOnClickListener(onClickListener);

cowImageButton.setOnTouchListener(new View.OnTouchListener() {

    public boolean onTouch(View v, MotionEvent event) {
        int eventAction = event.getAction();
        if (eventAction == MotionEvent.ACTION_UP) {
            // Отпускаем палец
            if (mStreamID > 0)
                mSoundPool.stop(mStreamID);
        }
        if (eventAction == MotionEvent.ACTION_DOWN) {
            // Нажимаем на кнопку
            mStreamID = playSound(mCowSound);
        }
        if (event.getAction() == MotionEvent.ACTION_CANCEL) {
            mSoundPool.stop(mStreamID);
        }
        return true;
    }
});
}

View.OnClickListener onClickListener = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.imageButtonCow:
                playSound(mCowSound);
                break;
            case R.id.imageButtonChicken:
                playSound(mChickenSound);
                break;
            case R.id.imageButtonCat:
                playSound(mCatSound);
                break;
            case R.id.imageButtonDuck:
                playSound(mDuckSound);
                break;
            case R.id.imageButtonSheep:
                playSound(mSheepSound);
                break;
            case R.id.imageButtonDog:
                playSound(mDogSound);
                break;
        }
    }
};

```

```

@TargetApi(Build.VERSION_CODES.LOLLIPOP)
private void createNewSoundPool() {
    AudioAttributes attributes = new AudioAttributes.Builder()
        .setUsage(AudioAttributes.USAGE_GAME)
        .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
        .build();
    mSoundPool = new SoundPool.Builder()
        .setAudioAttributes(attributes)
        .build();
}

@SuppressWarnings("deprecation")
private void createOldSoundPool() {
    mSoundPool = new SoundPool(3, AudioManager.STREAM_MUSIC, 0);
}

private int playSound(int sound) {
    if (sound > 0) {
        mStreamID = mSoundPool.play(sound, 1, 1, 1, 0, 1);
    }
    return mStreamID;
}

private int loadSound(String fileName) {
    AssetFileDescriptor afd;
    try {
        afd = mAssetManager.openFd(fileName);
    } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText(getApplicationContext(), "Не могу загрузить файл " + fileName,
            Toast.LENGTH_SHORT).show();
        return -1;
    }
    return mSoundPool.load(afd, 1);
}

@Override
protected void onResume() {
    super.onResume();

    if (android.os.Build.VERSION.SDK_INT < Build.VERSION_CODES.LOLLIPOP) {
        // Для устройств до Android 5
        createOldSoundPool();
    } else {
        // Для новых устройств
        createNewSoundPool();
    }

    mAssetManager = getAssets();

    // получим идентификаторы
    mCatSound = loadSound("cat.ogg");
    mChickenSound = loadSound("chicken.ogg");
    mCowSound = loadSound("cow.ogg");
    mDogSound = loadSound("dog.ogg");
    mDuckSound = loadSound("duck.ogg");
    mSheepSound = loadSound("sheep.ogg");
}

```

```
    }

    @Override
    protected void onPause() {
        super.onPause();
        mSoundPool.release();
        mSoundPool = null;
    }
}
```

При загрузке файлов метод **load()** возвращает идентификатор **soundID**, который сохраняем для дальнейшего использования. Объявим для каждого звука отдельную переменную, если же звуков много лучше завести для этого ассоциативный массив.

Файловый дескриптор **AssetFileDescriptor** для файла из директории **assets** получаем с помощью метода **openFd()**, принимающего в качестве параметра имя файла. Если файл не найден или не может быть открыт, то выводим сообщение и в качестве **soundID** возвращаем -1.

По нажатию кнопки вызываем метод **playSound()**, передавая ему нужный идентификатор звука. В методе проверяем этот идентификатор. Если файл не был найден, то метод **loadSound()** возвращает -1, а если метод **load()** класса **SoundPool** не смог загрузить файл, то **soundID** будет равен 0, поэтому проверяем, что *SoundID* > 0, что означает, что файл был успешно загружен. Если же все хорошо, то вызываем метод **play()**.

В версии Android 5.0 конструктор класса **SoundPool** является устаревшим. В коде использовано условие **if** с проверкой версии системы на устройстве, а также использованы аннотации, чтобы студия не ругалась на устаревший метод. Про аннотации мы поговорим в другой статье, пока воспринимайте их как подсказку-предупреждение при написании кода, чтобы выбрать правильный вариант.

Программа держит загруженные звуки в памяти. Если они вам не нужны, то нужно освободить ресурсы. Я сделал это в методе **onPause()**, соответственно загрузку пришлось перенести в **onResume()**.

Запустим программу и выясним, так кто-же сказал Мяу?



Написано по мотивам статьи Звуковые эффекты (<http://ru-code-android.blogspot.ru/2013/03/blog-post.html>).

Один из читателей захотел выводить звук не через щелчок, а нажатие на кнопку. А когда палец открывается от экрана, то звук должен прекращаться. Получился интересный эффект, который мы нашли сообща. Код для кнопки с коровой (предыдущий код лучше убрать):

```
cowImageButton.setOnTouchListener(new View.OnTouchListener() {

    public boolean onTouch(View v, MotionEvent event) {
        int eventAction = event.getAction();
        if (eventAction == MotionEvent.ACTION_UP) {
            // Отпускаем палец
            if (mStreamID > 0)
                mSoundPool.stop(mStreamID);
        }
        if (eventAction == MotionEvent.ACTION_DOWN) {
            // Нажимаем на кнопку
            mStreamID = playSound(mCowSound);
        }
        if (event.getAction() == MotionEvent.ACTION_CANCEL) {
            mSoundPool.stop(mStreamID);
        }
        return true;
    }
});
```

При воспроизведении звука мы получаем его идентификатор, используемый для остановки воспроизведения.