

توسعه برنامه های

موبایل

جلسه دهم مجازی

بخش دوم

سحر صادقی

## آموزش به دست آوردن لیست Sensor های فعال در برنامه نویسی اندروید

در این سری از آموزش برنامه نویسی اندروید به آموزش به دست آوردن لیست Sensor های فعال در برنامه نویسی اندروید می پردازیم این آموزش کاربردی است در ادامه با ما همراه باشید. در این آموزش لیست تمامی سنسور ها در ListView نمایش داده می شود.

پس در activity\_main.xml یک ListView همانند زیر قرار دهید.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context="MainActivity"
8   android:background="#ECEFF1">
9
10  <ListView
11    android:layout_width="match_parent"
12    android:layout_height="match_parent"
13    android:layout_alignParentTop="true"
14    android:layout_centerHorizontal="true"
15    android:id="@+id/listview1"/>
16
17 </RelativeLayout>
```

سپس کد مربوط به MainActivity.java هم همانند زیر می شود.

```
1 package com.android_examples.sensorslist_android_examplescom;
2 import android.content.Context;
3 import android.hardware.Sensor;
4 import android.hardware.SensorManager;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.widget.ArrayAdapter;
8 import android.widget.ListView;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class MainActivity extends AppCompatActivity {
13
14     ListView listView ;
```

```

15  SensorManager sensorManager ;
16  List<Sensor> listsensor;
17  List<String> liststring ;
18  ArrayAdapter<String> adapter ;
19
20  @Override
21  protected void onCreate(Bundle savedInstanceState) {
22      super.onCreate(savedInstanceState);
23      setContentView(R.layout.activity_main);
24
25      listView = (ListView)findViewById(R.id.listview1);
26
27      liststring = new ArrayList<String>();
28
29      sensorManager =
30  (SensorManager) getSystemService(Context.SENSOR_SERVICE);
31
32      listsensor = sensorManager.getSensorList(Sensor.TYPE_ALL);
33
34      for(int i=0; i<listsensor.size(); i++){
35
36          liststring.add(listsensor.get(i).getName());
37      }
38
39      adapter = new ArrayAdapter<String>(MainActivity.this,
40          android.R.layout.simple_list_item_2,
41          android.R.id.text1, liststring
42      );
43
44      listView.setAdapter(adapter);
45
46  }

```

در بالا از SensorManager استفاده کردیم و یک لیست درست کردیم و با استفاده از یک حلقه نام تمامی انها را در لیست اضافه می کنیم و سپس در ListView نمایش میدهیم

## تشخیص لرزش گوشی در اندروید

میخواهیم به تشخیص لرزش گوشی در اندروید یا shake در اندروید بپردازیم. منظور از shake و لرزش در اندروید هنگامی است که گوشی خود را تکان می دهید شاید بارها در برنامه های مختلف دیده باشید که با تکان دادن گوشی عملیاتی انجام داده میشود. برای اینکار روش های مختلفی را داریم و ما در اینجا به روشی میپردازیم که نسبتا جدید و دقیق تر است.

قبل از شروع ابتدا اجازه دسترسی به برنامه‌ها می‌دهیم :

```
1 <uses-permission  
1 android:name="android.permission.READ_PHONE_STATE"></uses-permission>
```

ما برای اینکار از AccelerometerListener استفاده میکنیم که باید آن را در قالب interface ایجاد کنیم و اکتیویتی اصلیمان را از این رابط implements میکنیم که در بخش کدهای MainActivity خواهید دید.

خب قبل از رفتن به اکتیویتی اصلی لازمه که ابتدا این interface را ایجاد کنید برای اینکار هنگام ایجاد یک فایل جاوا interface را انتخاب کنید سپس کدهای زیر را در آن بنویسید در اینجا ما نام فایل را AccelerometerListener قرار داده‌ایم :

```
1 public interface AccelerometerListener {  
2     public void onAccelerationChanged(float x, float y, float z);  
3  
4     public void onShake(float force);  
5 }
```

حالا یک کلاس ایجاد کنید و کدهای زیر را در آن قرار دهید نام این کلاس در اینجا accelerometerManager می‌باشد:

```
1 import android.content.Context;  
2 import android.hardware.Sensor;  
3 import android.hardware.SensorEvent;  
4 import android.hardware.SensorEventListener;  
5 import android.hardware.SensorManager;  
6 import android.widget.Toast;  
7  
8 import java.util.List;  
9  
10 /**  
11  * Created by Sepehr on 18/07/2017.  
12  */  
13  
14 public class AccelerometerManager {  
15     private static Context aContext=null;  
16  
17  
18     /** Accuracy configuration */  
19     private static float threshold = 15.0f;
```

```

20 private static int interval    = 200;
21
22 private static Sensor sensor;
23 private static SensorManager sensorManager;
24 // you could use an OrientationListener array instead
25 // if you plans to use more than one listener
26 private static AccelerometerListener listener;
27
28 /** indicates whether or not Accelerometer Sensor is supported */
29 private static Boolean supported;
30 /** indicates whether or not Accelerometer Sensor is running */
31 private static boolean running = false;
32
33 /**
34  * Returns true if the manager is listening to orientation changes
35  */
36 public static boolean isListening() {
37     return running;
38 }
39
40 /**
41  * Unregisters listeners
42  */
43 public static void stopListening() {
44     running = false;
45     try {
46         if (sensorManager != null && sensorEventListener != null) {
47             sensorManager.unregisterListener(sensorEventListener);
48         }
49     } catch (Exception e) {}
50 }
51
52 /**
53  * Returns true if at least one Accelerometer sensor is available
54  */
55 public static boolean isSupported(Context context) {
56     aContext = context;
57     if (supported == null) {
58         if (aContext != null) {
59
60
61             sensorManager = (SensorManager) aContext.
62                 getSystemService(Context.SENSOR_SERVICE);
63
64             // Get all sensors in device
65             List<Sensor> sensors = sensorManager.getSensorList(
66                 Sensor.TYPE_ACCELEROMETER);
67
68             supported = new Boolean(sensors.size() > 0);

```

```

69
70
71
72     } else {
73         supported = Boolean.FALSE;
74     }
75 }
76 return supported;
77 }
78
79 /**
80  * Configure the listener for shaking
81  * @param threshold
82  *     minimum acceleration variation for considering shaking
83  * @param interval
84  *     minimum interval between to shake events
85  */
86 public static void configure(int threshold, int interval) {
87     AccelerometerManager.threshold = threshold;
88     AccelerometerManager.interval = interval;
89 }
90
91 /**
92  * Registers a listener and start listening
93  * @param accelerometerListener
94  *     callback for accelerometer events
95  */
96 public static void startListening( AccelerometerListener accelerometerListener )
97 {
98
99     sensorManager = (SensorManager) aContext.
100         getSystemService(Context.SENSOR_SERVICE);
101
102     // Take all sensors in device
103     List<Sensor> sensors = sensorManager.getSensorList(
104         Sensor.TYPE_ACCELEROMETER);
105
106     if (sensors.size() > 0) {
107
108         sensor = sensors.get(0);
109
110         // Register Accelerometer Listener
111         running = sensorManager.registerListener(
112             sensorEventListener, sensor,
113             SensorManager.SENSOR_DELAY_GAME);
114
115         listener = accelerometerListener;
116     }
117

```

```

118
119 }
120
121 /**
122  * Configures threshold and interval
123  * And registers a listener and start listening
124  * @param accelerometerListener
125  *      callback for accelerometer events
126  * @param threshold
127  *      minimum acceleration variation for considering shaking
128  * @param interval
129  *      minimum interval between to shake events
130  */
131 public static void startListening(
132     AccelerometerListener accelerometerListener,
133     int threshold, int interval) {
134     configure(threshold, interval);
135     startListening(accelerometerListener);
136 }
137
138 /**
139  * The listener that listen to events from the accelerometer listener
140  */
141 private static SensorEventListener sensorEventListener =
142     new SensorEventListener() {
143
144         private long now = 0;
145         private long timeDiff = 0;
146         private long lastUpdate = 0;
147         private long lastShake = 0;
148
149         private float x = 0;
150         private float y = 0;
151         private float z = 0;
152         private float lastX = 0;
153         private float lastY = 0;
154         private float lastZ = 0;
155         private float force = 0;
156
157         public void onAccuracyChanged(Sensor sensor, int accuracy) {}
158
159         public void onSensorChanged(SensorEvent event) {
160             // use the event timestamp as reference
161             // so the manager precision won't depends
162             // on the AccelerometerListener implementation
163             // processing time
164             now = event.timestamp;
165
166             x = event.values[0];

```

```

167     y = event.values[1];
168     z = event.values[2];
169
170     // if not interesting in shake events
171     // just remove the whole if then else block
172     if (lastUpdate == 0) {
173         lastUpdate = now;
174         lastShake = now;
175         lastX = x;
176         lastY = y;
177         lastZ = z;
178         Toast.makeText(aContext,"No Motion detected",
179             Toast.LENGTH_SHORT).show();
180
181     } else {
182         timeDiff = now - lastUpdate;
183
184         if (timeDiff > 0) {
185
186             /*force = Math.abs(x + y + z - lastX - lastY - lastZ)
187              / timeDiff;*/
188             force = Math.abs(x + y + z - lastX - lastY - lastZ);
189
190             if (Float.compare(force, threshold) > 0 ) {
191                 //Toast.makeText(Accelerometer.getContext(),
192                 //(now-lastShake)+" >= "+interval, 1000).show();
193
194                 if (now - lastShake >= interval) {
195
196                     // trigger shake event
197                     listener.onShake(force);
198                 }
199                 else
200                 {
201                     Toast.makeText(aContext,"No Motion detected",
202                         Toast.LENGTH_SHORT).show();
203
204                 }
205                 lastShake = now;
206             }
207             lastX = x;
208             lastY = y;
209             lastZ = z;
210             lastUpdate = now;
211         }
212         else
213         {
214             Toast.makeText(aContext,"No Motion detected",
215 Toast.LENGTH_SHORT).show();

```



```

216
217     }
218 }
219 // trigger change event
220 listener.onAccelerationChanged(x, y, z);
221 }
222
223 };
224
    }

```

خب امیدوارم با دیدن این کد ها ناامید نشده باشید. در اینجا بخش اصلی کار ما در متد `ensorEventListener` انجام می شود که با تعریف تعدادی متغیر همراه است که وضعیت لرزش یا عدم لرزش به کمک آن ها انجام می شود. حالا میرسیم به اکتیویتی اصلیمان همانطور که قبلا گفتیم باید از رابط `AccelerometerListener` استفاده کنیم برای اینکار بصورت زیر عمل میکنیم و بقیه کد ها را هم مینویسیم:

```

1  public class MainActivity extends AppCompatActivity implements
2  AccelerometerListener {
3
4
5      @Override
6      protected void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_main);
9
10
11     }
12     public void onAccelerationChanged(float x, float y, float z) {
13         // TODO Auto-generated method stub
14
15     }
16
17     public void onShake(float force) {
18
19         // Do your stuff here
20
21         // Called when Motion Detected
22         Toast.makeText(getBaseContext(), "Motion detected",
23             Toast.LENGTH_SHORT).show();
24
25     }
26
27     @Override
28     public void onResume() {
29         super.onResume();
30         Toast.makeText(getBaseContext(), "onResume Accelerometer Started",

```

```

31         Toast.LENGTH_SHORT).show();
32
33         //Check device supported Accelerometer sensor or not
34         if (AccelerometerManager.isSupported(this)) {
35
36             //Start Accelerometer Listening
37             AccelerometerManager.startListening(this);
38         }
39     }
40
41     @Override
42     public void onStop() {
43         super.onStop();
44
45         //Check device supported Accelerometer sensor or not
46         if (AccelerometerManager.isListening()) {
47
48             //Start Accelerometer Listening
49             AccelerometerManager.stopListening();
50
51             Toast.makeText(getBaseContext(), "onStop Accelerometer Stopped",
52                 Toast.LENGTH_SHORT).show();
53         }
54
55     }
56
57     @Override
58     public void onDestroy() {
59         super.onDestroy();
60         Log.i("Sensor", "Service distroy");
61
62         //Check device supported Accelerometer sensor or not
63         if (AccelerometerManager.isListening()) {
64
65             //Start Accelerometer Listening
66             AccelerometerManager.stopListening();
67
68             Toast.makeText(getBaseContext(), "onDestroy Accelerometer Stopped",
69                 Toast.LENGTH_SHORT).show();
70         }
71
72     }
73 }

```

خب همانطور که قبلا رابطی با نام AccelerometerListener ایجاد کردیم اکتیویتی اصلیمان را از ان implements کردیم. سپس از متد های onResume و onShake و onStop و onDestroy استفاده کردیم که با سه تای ان ها اشنایی پیدا کردیم در بحث اکتیویتی ها و متد onShake مبحث جدید میباشد که همانطور که از toast ان متوجه هستید هنگام تکان دادن گوشی عملیات داخل ان انجام می شود.