

توسعه برنامه های

موبایل

جلسه دهم مجازی

بخش اول

سحر صادقی

## پیاده سازی کلاس SensorManager و دسترسی به حسگرهای دستگاه اندروید

برای دسترسی به حسگرهای دستگاه اندروید می بایست نمونه ای از کلاس SensorManager را پیاده سازی نمایید. از جمله ی این حسگرها می توان به شتاب سنج یا accelerometer اشاره کرد. متأسفانه، امکان تست شتاب سنج و شبیه سازی آن در Android emulator وجود ندارد.

برای دسترسی به کلاس SensorManager کافی است متد `getSystemService(SENSOR_SERVICE)` را فراخوانی نمایید. کلاس Sensor ثوابت متعددی (constants) برای دسترسی به حسگرهای مختلف فراهم می آورد.

- `Sensor.TYPE_GYROSCOPE`
- `Sensor.TYPE_MAGNETIC_FIELD`
- `Sensor.TYPE_ORIENTATION`
- `Sensor.TYPE_ACCELEROMETER`

می توانید با فراخوانی متد `sensorManager.getDefaultSensor()` به حسگر مربوطه دسترسی داشته باشید. این متد نوع حسگر (sensor type) و میزان زمان غیر فعال بودن تا فراخوانی حسگر را که به عنوان ثابت در سطح کلاس SensorManager تعریف شده، به عنوان پارامتر می پذیرد.

## گوش فرادهنده به تغییرات حسگر (Sensor listener)

پس از دسترسی به حسگر مربوطه، لازم است یک `SensorEventListener` برای آن تعریف نمایید. این listener به تغییرات مرتبط با حسگر (sensor) گوش می دهد و به محض رخداد تغییر در اطلاعات حسگر مورد نظر، از آن با خبر می شود.

به منظور جلوگیری از اتلاف باتری، می توانید listener خود را در متد `onResume()` اعلان و ثبت نموده و داخل متد `onPause()` از حالت مزبور خارج نمایید (deregister کنید).

## آموزش استفاده ی کاربردی از حسگر Accelerometer

در این بخش یک اپلیکیشن خواهیم نوشت که در صورت تکان خوردن دستگاه، به صورت خودکار رنگ پس زمینه را تغییر می دهد. برای این منظور ابتدا یک پروژه ی اندرویدی و activity به ترتیب به نام های `de.vogella.android.sensor` و `SensorTestActivity` ایجاد نمایید.

محتوای فایل layout خود را به صورت زیر ویرایش نمایید.

```

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

<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
1  android:layout_width="match_parent"
2  android:layout_height="match_parent"
  android:orientation="vertical">
3  <TextView android:id="@+id/textView"
4  android:layout_width="match_parent"
  android:layout_height="match_parent" android:text="Shake to get a
  toast and to switch color">

</TextView></LinearLayout>

```

حال کد کلاسی activity خود را به صورت زیر ویرایش نمایید.

?

```

package de.vogella.android.sensor;

import android.app.Activity;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Toast;

2 public class SensorTestActivity extends Activity implements SensorEventListener {
3     private SensorManager sensorManager;
4     private boolean color = false;

```

```

5 privateView view;
6 private long lastUpdate;
7 /** Called when the activity is first created. */
8 @Override
9 public void onCreate(Bundle savedInstanceState) {
10     requestWindowFeature(Window.FEATURE_NO_TITLE);
11     getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
12         WindowManager.LayoutParams.FLAG_FULLSCREEN);
13     super.onCreate(savedInstanceState);
14     setContentView(R.layout.main);
15     view = findViewById(R.id.textView);
16     view.setBackgroundColor(Color.GREEN);
17     sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
18     lastUpdate = System.currentTimeMillis();
19 }
20 @Override
21 public void onSensorChanged(SensorEvent event) {
22     if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
23         getAccelerometer(event);
24     }
25 }
26 private void getAccelerometer(SensorEvent event) {
27     float[] values = event.values;
28     // Movement
29     float x = values[0];
30     float y = values[1];
31     float z = values[2];
32     float accelerationSquareRoot = (x * x + y * y + z * z)
33         / (SensorManager.GRAVITY_EARTH *
34 SensorManager.GRAVITY_EARTH);
35     long actualTime = event.timestamp;
36     if (accelerationSquareRoot >= 2) //

```

```

16         {
17             if (actualTime - lastUpdate < 200) {
18                 return;
19             }
20             lastUpdate = actualTime;
21             Toast.makeText(this, "Device was shuffled", Toast.LENGTH_SHORT)
22                 .show();
23             if (color) {
24                 view.setBackgroundColor(Color.GREEN);
25             } else {
26                 view.setBackgroundColor(Color.RED);
27             }
28             color = !color;
29         }
30     }
31     @Override
32     public void onAccuracyChanged(Sensor sensor, int accuracy) {
33     }
34     @Override
35     protected void onResume() {
36         super.onResume();
37         // register this class as a listener for the orientation and
38         // accelerometer sensors
39         sensorManager.registerListener(this,
40             sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
41             SensorManager.SENSOR_DELAY_NORMAL);
42     }
43     @Override
44     protected void onPause() {
45         // unregister listener
46         super.onPause();
47         sensorManager.unregisterListener(this);
48     }

```

77 }

78 }

79

80

## آموزش کاربردی: ساخت یک قطب نما

یک پروژه و activity جدید اندرویدی به ترتیب به نام های `de.vogella.android.sensor.compass` و `MainActivity` ایجاد نمایید.

اکنون یک کلاس `View` با پیاده سازی اختصاصی به صورت زیر ایجاد کنید.

[?](#)

```
1 package de.vogella.android.sensor.compass;
2 import android.content.Context;
3 import android.graphics.Canvas;
4 import android.graphics.Color;
5 import android.graphics.Paint;
6 import android.view.View;
7 public class MyCompassView extends View {
8     private Paint paint;
9     private float position = 0;
10    public MyCompassView(Context context) {
11        super(context);
12        init();
13    }
14    private void init() {
15        paint = new Paint();
16        paint.setAntiAlias(true);
17        paint.setStrokeWidth(2);
18        paint.setTextSize(25);
19        paint.setStyle(Paint.Style.STROKE);
20        paint.setColor(Color.WHITE);
21    }
```

```

20         @Override
21         protected void onDraw(Canvas canvas) {
22             int xPoint = getMeasuredWidth() / 2;
23             int yPoint = getMeasuredHeight() / 2;
24             float radius = (float) (Math.max(xPoint,
25 yPoint) * 0.6);
26             canvas.drawCircle(xPoint, yPoint, radius,
27 paint);
28             canvas.drawRect(0, 0, getMeasuredWidth(),
29 getMeasuredHeight(), paint);
30             // 3.143 is a good approximation for the
31 circle
32             canvas.drawLine(
33                 xPoint,
34                 yPoint,
35                 (float) (xPoint + radius
36 *
37 Math.sin((double) (-position) / 180 * 3.143)),
38                 (float) (yPoint - radius
39 *
40 Math.cos((double) (-position) / 180 * 3.143)), paint);
41             canvas.drawText(String.valueOf(position),
42 xPoint, yPoint, paint);
43         }
44     }
45     public void updateData(float position) {
46         this.position = position;
47         invalidate();
48     }
49 }

```

**بدنه ی کلاس activity پروژه ی خود را به صورت زیر ویرایش کنید.**

[?](#)

```

1 package de.vogella.android.sensor.compass;
2 import android.app.Activity;
3 import android.content.Context;

```

```

3 import android.hardware.Sensor;
4 import android.hardware.SensorEvent;
5 import android.hardware.SensorEventListener;
6 import android.hardware.SensorManager;
7 import android.os.Bundle;
8 import android.util.Log;
9 import android.widget.Toast;
10 public class MainActivity extends Activity {
11     private static SensorManager sensorService;
12     private MyCompassView compassView;
13     private Sensor sensor;
14     /** Called when the activity is first created. */
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         compassView = new MyCompassView(this);
19         setContentView(compassView);
20         sensorService = (SensorManager)
21 getSystemService(Context.SENSOR_SERVICE);
22         sensor =
23 sensorService.getDefaultSensor(Sensor.TYPE_ORIENTATION);
24         if (sensor != null) {
25             sensorService.registerListener(mySensorEventListener, sensor,
26                                     SensorManager.SENSOR_DELAY
27 _NORMAL);
28             Log.i("Compass MainActivity",
29 "Registered for ORIENTATION Sensor");
30         } else {
31             Log.e("Compass MainActivity",
32 "Registered for ORIENTATION Sensor");
33             Toast.makeText(this, "ORIENTATION Sensor
34 not found",
35                                     Toast.LENGTH_LONG).show();
36             finish();
37 }

```



```

2          }
3      }

2      private SensorEventListener mySensorEventListener = new
4 SensorEventListener() {

2          @Override
5          public void onAccuracyChanged(Sensor sensor, int
2 accuracy) {
6          }

2          @Override
7          public void onSensorChanged(SensorEvent event) {

2              // angle between the magnetic north
8 direction

2              // 0=North, 90=East, 180=South, 270=West
9              float azimuth = event.values[0];
3              compassView.updateData(azimuth);
0          }

3      };
1      @Override
3      protected void onDestroy() {
2          super.onDestroy();
3          if (sensor != null) {
3              sensorService.unregisterListener(mySensorE
3 ventListener);
4          }

3      }
5 }
3
6
3
7
3
8
3
9

```

4  
0  
4  
1  
4  
2  
4  
3  
4  
4  
4  
5  
4  
6  
4  
7  
4  
8  
4  
9  
5  
0  
5  
1  
5  
2  
5  
3

