



Aplikacje mobilne

Czujniki

dr Tomasz Jach
Instytut Informatyki,
Uniwersytet Śląski



Uprawnienia

W Androidzie aby korzystać z większości elementów, należy poprosić system o dane uprawnienie.

Zmieniamy je w pliku **AndroidManifest.xml**

Listę uprawnień można znaleźć tutaj:

<http://developer.android.com/reference/android/Manifest.permission.html>

```
Java - Horoskop/AndroidManifest.xml - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Package Explo...
Horoskop
  Android 4.4.2
  Android Private Libraries
  src
  gen [Generated Java File]
  assets
  bin
  libs
  res
  AndroidManifest.xml
  ic_launcher-web.png
  proguard-project.txt
  project.properties
MainActivity.java dimens.xml main.xml activity_main.xml *Horoskop Ma
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.tjach.horoskop"

  android:versionCode="1"
  android:versionName="1.0" >
  <uses-permission android:name="android.permission.RECEIVE_SMS" />
  <uses-sdk
    android:minSdkVersion="19"
    android:targetSdkVersion="19" />
  <uses-permission />

  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
      android:name=".MainActivity"
      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>
Manifest Application Permissions Instrumentation AndroidManifest.xml
```



Czujniki 101

```
1 package com.tjach.sensors;
2
3 import java.util.List;
13
14 public class MainActivity extends Activity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         SensorManager manager = (SensorManager) getSystemService(SENSOR_SERVICE);
22         List<Sensor> listaSensorow = manager.getSensorList(Sensor.TYPE_ALL);
23         for(int i=0;i<listaSensorow.size();i++) {
24             Log.d("Sensory", listaSensorow.get(i).getName());
25         }
26     }
27 }
```

Linia 21 i 22: sprawdzamy jakie mamy sensory dostępne w urządzeniu.

Linia 24: do LogCata wypisujemy dostępne sensory (w 22 linijce możemy filtrować po typie, np. tylko sensory lokalizacyjne, albo temperatury).

Odczytujemy orientację telefonu



1. Musimy zaimplementować interfejs „SensorEventListener”. Dzięki temu możliwe będzie podpięcie się pod wartości sensorów.
2. Przydadzą się różne zmienne (korzystamy z akcelerometru i magnetometru)
3. Pamiętaj o uprawnieniach.

```
dimens.xml | MainActivity... | activity_mai... | Sensors Mani... | Sensors Mani... | 2
1 package com.tjach.sensors;
2
3+ import android.app.Activity;
14
15 public class MainActivity extends Activity implements SensorEventListener {
16
17- private
18     Sensor accelerometer;
19     Sensor magnetometer;
20     SensorManager mSensorManager;
21     float[] mGravity;
22     float[] mGeomagnetic;
23     TextView azymut;
24     TextView przechyl1;
25     TextView przechyl2;
26
```

```
7
8
9
10
11
12
<uses-sdk
    android:minSdkVersion="19"
    android:targetSdkVersion="19" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Odczytujemy orientację telefonu



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

    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    accelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    magnetometer = mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    azymut = (TextView) findViewById(R.id.textView1);
    przechyl1 = (TextView) findViewById(R.id.textView2);
    przechyl2 = (TextView) findViewById(R.id.textView3);
}
}
```

Za pomocą menagera sensorów podpinamy się pod dwa sensory – akcelerometr i czujnik pola magnetycznego. Przypisuję też TextView na formie do zmiennych w programie.

Odczytujemy orientację telefonu



```
39     }
40     protected void onResume() {
41         super.onResume();
42         mSensorManager.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_UI);
43         mSensorManager.registerListener(this, magnetometer, SensorManager.SENSOR_DELAY_UI);
44     }
45
46     protected void onPause() {
47         super.onPause();
48         mSensorManager.unregisterListener(this);
49     }
50
51     public void onAccuracyChanged(Sensor sensor, int accuracy) { }
52
```

W metodzie `onResume()` rejestruję aplikację jako kroszystającą z czujników. W metodzie `onPause()` – wyrejestrowuję.

Z racji interfejsu trzeba zaimplementować metody **`onAccuracyChanged()`** oraz **`onSensorChanged()`**.

Pierwsza wywołuje się jeśli z jakichś powodów zmieni się dokładność sensora, druga – gdy zmieni się jego wartość.

Odczytujemy orientację telefonu



```
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
        mGravity = event.values;
    if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
        mGeomagnetic = event.values;
    if (mGravity != null && mGeomagnetic != null) {
        float R[] = new float[9];
        float I[] = new float[9];
        boolean success = SensorManager.getRotationMatrix(R, I, mGravity, mGeomagnetic);
        if (success) {
            float orientation[] = new float[3];
            SensorManager.getOrientation(R, orientation);
            azymut.setText(String.valueOf(orientation[0])); // orientation contains: azymut, pitch and ro
            przechyl1.setText(String.valueOf(orientation[1]));
            przechyl2.setText(String.valueOf(orientation[2]));
        }
    }
}
```

Właściwe odczytanie orientacji przez funkcję `getRotationMatrix()` i `getOrientation()`.

Uwzględnia ona dane z czujników i zwraca azymut, przechył w poziomie i przechył w pionie.

Ćwiczenie 1



- Napisz prosty program typu poziomicca. Zadbaj o prawidłowe wygładzanie wartości (wartości czujników bardzo szybko się zmieniają)
- Odczytuj informacje co jakiś czas, a nie co ich zmianę

Ćwiczenie 2



- Napisz program typu labirynt. Za pomocą czujnika orientacji i czasu (sprawdź jak mierzyć czas w Androidzie) napisz program typu: „Idź przez 10 sekund na południowy wschód, potem 20 sekund na północ”.
- Jeśli chcesz i jesteś w stanie – sprawdzaj liczbę kroków (trudne!) za pomocą skokowych zmian akcelerometru.



Czujnik zbliżeniowy

- Zasada działania identyczna, zmienia się tylko rodzaj wykorzystywanego czujnika:

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

    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    // accelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    // magnetometer = mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    proximity = mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
    azymut = (TextView) findViewById(R.id.textView1);
    przechyl1 = (TextView) findViewById(R.id.textView2);
    przechyl2 = (TextView) findViewById(R.id.textView3);
}
protected void onResume() {
    super.onResume();
    //mSensorManager.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_UI);
    //mSensorManager.registerListener(this, magnetometer, SensorManager.SENSOR_DELAY_UI);
    mSensorManager.registerListener(this, proximity, SensorManager.SENSOR_DELAY_UI);
}
public void onSensorChanged(SensorEvent event) {
    azymut.setText(String.valueOf(event.values[0]));
}
```

Ćwiczenie 3



- Napisz aplikację, która wyświetla użytkownikowi spokojną grafikę dopóki ten nie przybliży ręki do czujnika. Wtedy następuje podmiana grafiki na coś straszego 😊

Ćwiczenie 4



- Napisz prostą galerię zdjęć, w której zdjęcia będą przewijane czujnikiem zbliżeniowym.
- Zakładamy, że każde zbliżenie ręki – następne zdjęcie.
- Skorzystaj w tym celu z komponentów omówionych na wykładzie [03 Widżety oparte na listach](#) (slajdy od 38).

Ćwiczenie 5



- Za pomocą czujnika światła (TYPE_LIGHT) określaj jak jasno jest w pokoju. Przerób swoją latarkę tak, aby dostosowywała jasność nie za pomocą slidera, ale automatycznie.
- Sprawdź jakie inne czujniki można wykorzystać:
http://developer.android.com/guide/topics/sensors/sensors_environment.html