

Programação Para Dispositivos Móveis I

SENSORES

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt

Índice

- Sensores;
- Leitura Adicional.

Sensores

- Os sensores são (tipicamente) componentes físicos que permitem perceber fenômenos externos a partir do dispositivo Android.
- São utilizados em aplicações que necessitam de contextos externos para funcionar, ou para adicionar recursos avançados (ex: localização ou movimento ao seu aplicativo);
- O seu uso é feito diretamente a partir de serviços de sistema sobre os quais podemos registar *callbacks* para serem notificados de eventuais alterações dos valores registados.

Sensores

A plataforma Android suporta várias categorias de sensores:

- **Movimento:** sensores que medem forças de aceleração, rotação e outros ao longo de três eixos (x,y,z). Nesta categoria incluem-se acelerómetros, sensores de gravidade, giroscópios, e sensores vetoriais de rotação.
- **Ambiente:** sensores que medem parâmetros ambientais, como temperatura, pressão do ar ambiente, iluminação e humidade. Esta categoria inclui barómetros, fotómetros e termómetros.
- **Posição:** sensores que medem a posição física de um dispositivo. Esta categoria inclui sensores de orientação e magnetómetros.

Sensores

Podemos aceder aos sensores disponíveis no dispositivo e adquirir dados em brutos usando a plataforma de sensores do Android. Fornece várias classes e interfaces que ajudam na execução de tarefas relacionadas com sensores:

- Determinar os sensores disponíveis no dispositivo;
- Determinar os recursos de um sensor individual:
 - Taxa de atualização;
 - Fabricante;
 - Requisitos de energia;
 - Resolução.
- Adquirir dados em brutos do sensor e definir a taxa de atualização mínima;
- Registrar e cancelar o registro de *listeners*;

Sensores

Sensor	Descrição	Usos comuns
TYPE_ACCELEROMETER	Mede a força de aceleração em m/s ² que é aplicada a um dispositivo nos três eixos físicos (x, y, e z), incluindo a força da gravidade.	Deteção de movimento (agitação, inclinação, etc.)
TYPE_AMBIENT_TEMPERATURE	Mede a temperatura ambiente em graus Celsius (°C).	Monitorização da temperatura do ar
TYPE_GRAVITY	Mede a força da gravidade em m/s ² que é aplicada a um dispositivo nos três eixos físicos (x, y, z).	Deteção de movimento (abandar, inclinar, etc.)
TYPE_GYROSCOPE	Mede a taxa de rotação de um dispositivo em rad/s em torno de cada um dos três eixos físicos (x, y, e z).	Deteção de rotação (girar, rodar, etc.)
TYPE_LIGHT	Mede o nível de luz ambiente (iluminação) em lx.	Controlar o brilho do ecrã
TYPE_LINEAR_ACCELERATION	Mede a força de aceleração em m/s ² que é aplicada a um dispositivo nos três eixos físicos (x, y, e z), excluindo a força da gravidade.	Monitorização da aceleração ao longo de um único eixo
TYPE_MAGNETIC_FIELD	Mede o campo geomagnético ambiente para os três eixos físicos (x, y, z) em μT .	Criação de uma bússola

Sensores

Sensor	Descrição	Usos comuns
TYPE_ORIENTATION	Mede os graus de rotação que um dispositivo faz em torno dos três eixos físicos (x, y, z). A partir do nível API 3 pode-se obter a matriz de inclinação e a matriz de rotação de um dispositivo usando o sensor de gravidade e o sensor de campo geomagnético em conjunto com o método <code>getRotationMatrix()</code> .	Determinação da posição do dispositivo.
TYPE_PRESSURE	Mede a pressão do ar ambiente em hPa ou mbar.	Monitorização de alterações de pressão de ar.
TYPE_PROXIMITY	Mede a proximidade de um objeto em cm em relação ao ecrã de visualização de um dispositivo. Este sensor é tipicamente utilizado para determinar se o dispositivo está a ser segurado junto ao ouvido de uma pessoa.	Posição do dispositivo durante uma chamada.
TYPE_RELATIVE_HUMIDITY	Mede a humidade ambiente relativa em percentagem (%).	Monitorização do ponto de orvalho, humidade absoluta e relativa.
TYPE_ROTATION_VECTOR	Mede a orientação de um dispositivo, fornecendo os três elementos do vetor de rotação do dispositivo.	Deteção de movimento e deteção de rotação.
TYPE_TEMPERATURE	Mede a temperatura do dispositivo em graus Celsius (°C). A implementação deste sensor varia entre dispositivos e este sensor foi substituído pelo sensor <code>TYPE_AMBIENT_TEMPERATURE</code> no nível API 14	Monitorização das temperaturas.

Sensores

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes
TYPE_AMBIENT_TEMPERATURE	Yes	n/a	n/a	n/a
TYPE_GRAVITY	Yes	Yes	n/a	n/a
TYPE_GYROSCOPE	Yes	Yes	n/a	n/a
TYPE_LIGHT	Yes	Yes	Yes	Yes
TYPE_LINEAR_ACCELERATION	Yes	Yes	n/a	n/a
TYPE_MAGNETIC_FIELD	Yes	Yes	Yes	Yes
TYPE_ORIENTATION	Yes	Yes	Yes	Yes
TYPE_PRESSURE	Yes	Yes	n/a	n/a
TYPE_PROXIMITY	Yes	Yes	Yes	Yes
TYPE_RELATIVE_HUMIDITY	Yes	n/a	n/a	n/a
TYPE_ROTATION_VECTOR	Yes	Yes	n/a	n/a
TYPE_TEMPERATURE	Yes	Yes	Yes	Yes

Sensores em Android

Sensores de Ambiente

Para utilizar os sensores devemos **registar *listeners*** para utilizar os dados fornecidos pela plataforma Android.

- Devemos garantir que os *listeners* são apenas utilizados durante as etapas relevantes do ciclo de vida dos componentes em Android (Activities, Fragments, ...);
- O uso criterioso da API `SensorEventListener` ajuda na conservação de energia dos equipamentos Android;

Sensores de Ambiente

Exemplo obtenção de um sensor

```
private SensorManager sensorManager;  
private Sensor sensor;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
    sensor = sensorManager.getDefaultSensor(Sensor.TYPE_AMBIENT_TEMPERATURE);  
}
```

Instanciação do sensor do tipo
Ambient_Temperature

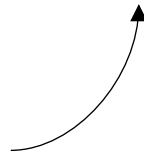


Sensores de Ambiente

Exemplo obtenção da lista dos sensores disponíveis

```
private SensorManager sensorManager;  
private List<Sensor> deviceSensors;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
    deviceSensors = sensorManager.getSensorList(Sensor.TYPE_ALL);  
}
```

Lista de todos os sensores
disponíveis no dispositivo




Sensores de Ambiente

Exemplo (Sensor de Luz Ambiente) [MainActivity.java]

```
public class MainActivity extends AppCompatActivity implements SensorEventListener {  
  
    private SensorManager sensorManager;  
    private Sensor luxSensor;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
        luxSensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);  
    }  
}
```

Implementação do
Listener de eventos



Obter sensor de Luz
Ambiente




Sensores de Ambiente

Exemplo (Sensor de Luz Ambiente) [MainActivity.java]

```
@Override
protected void onResume() {
    super.onResume();
    sensorManager.registerListener(this, luxSensor, SensorManager.SENSOR_DELAY_NORMAL);
}
```

Registrar o Listener respeitando o ciclo de vida da Activity



```
@Override
protected void onPause() {
    super.onPause();
    sensorManager.unregisterListener(this);
}
```

Remover o registo do listener



Sensores de Ambiente

Exemplo (Sensor de Luz Ambiente) [MainActivity.java]

Invocado cada vez que o valor registado pelo sensor sofre alteração

```
@Override  
public void onSensorChanged(SensorEvent event) {  
    Log.d("DEVICE_SENSOR", String.valueOf(event.values[0]));  
}
```

Obtenção do valor atual do sensor

```
@Override  
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
    //TODO: Implementar comportamento caso ocorra alguma alteração  
           no valor de accuracy do sensor  
}
```

Sensores em Android

Movimento

O sensor de movimento significativo aciona um evento cada vez que um movimento significativo é detetado e, em seguida, ele é desativado.

Uma moção significativa é uma moção que pode levar a uma alteração na localização do utilizador:

- Caminhar;
- Correr;
- Andar de bicicleta;
- Andar de carro;
- Etc.

Sensores de Movimento

Significant Motion Sensor [MainActivity.java]

```
private SensorManager sensorManager;  
private Sensor sensor;  
private TriggerEventListener triggerEventListener;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
    sensor = sensorManager.getDefaultSensor(Sensor.TYPE_SIGNIFICANT_MOTION);  
  
    triggerEventListener = new TriggerEventListener() {  
        @Override  
        public void onTrigger(TriggerEvent event) {  
            //TODO: Implementar comportamento caso ocorra um evento de SIGNIFICANT_MOTION  
        }  
    };  
    sensorManager.requestTriggerSensor(triggerEventListener, sensor);  
}
```


Sensores de Movimento

Step Counter Sensor [MainActivity.java]

```
private SensorManager sensorManager;  
private Sensor sensor;
```

@Override

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

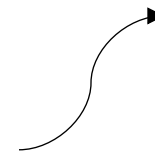
```
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
    sensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER);
```

```
}
```

Este sensor tem um valor maior de latência (10 segundos), mas uma accuracy superior ao TYPE_STEP_DETECTOR

Nota: deve ser declarada a permissão

ACTIVITY_RECOGNITION para que a aplicação use este sensor em dispositivos com Android 10 (API 29) ou superior.



Sensores em Android

Existem ainda em alguns dispositivos Android sensores especiais que requerem bibliotecas fornecidas diretamente a partir do fabricantes dos dispositivos.

- Exemplos de sensores não disponíveis na API Android:
 - Project Soli – Pixel 4 e Pixel XL;
- Sensores como o sensor de impressões digitais, requerem também permissões e usos especiais:
 - Sensibilidade dos dados previne o acesso aos dados em bruto;
 - Necessidade de proteger privacidade dos utilizadores.

Podemos no entanto usar os sensores biométricos em casos de uso previstos pelo Android como por exemplo autenticação pessoal

<https://developer.android.com/training/sign-in/biometric-auth>

Leitura Recomendada

Sensores:

https://developer.android.com/guide/topics/sensors/sensors_overview

Programação Para Dispositivos Móveis I

SENSORES

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt