Android — Sensores e Notificações

Flávio Velloso Laper

Universidade Fumec

5 de novembro de 2015

Programa da aula

Conteúdo

Sensores

- Dispositivos de hardware que fazem medições do ambiente físico.
- Exemplos:
  - Acelerômetro de três eixos.
  - Sensor de campo magnético de três eixos.
  - Sensor de temperatura.
  - Sensor de proximidade.
  - Sensor de orientação.
  - Sensor de luz.

SensorManager

- Serviço do sistema que gerencia os sensores do dispositivo.
- Obtenção de instância:
  - *getSystemService(Context.SENSOR_SERVICE)*.
- Acesso a sensor específico:
  - *SensorManager.getDefaultSensor(int type)*.

Alguns tipos de sensores

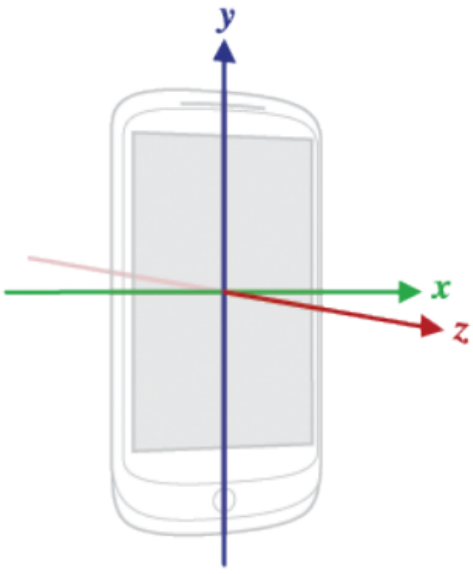
- TYPE_ACCELEROMETER
- TYPE_GRAVITY
- TYPE_GYROSCOPE
- TYPE_LIGHT
- TYPE_MAGNETIC_FIELD
- TYPE_PRESSURE
- TYPE_PROXIMITY
- TYPE_TEMPERATURE

SensorEvent

- Representa um evento de sensor.
- Manipula dados específicos do sensor:
  - tipo, *timestamp*, precisão, dados medidos, etc.

Sistema de coordenadas

- Os eixos de coordenadas do dispositivo (quando colocado sobre uma superfície) são:
  - X: deslocamento horizontal;
  - Y: deslocamente vertical;
  - Z: deslocamento em profundidade.
- O sistema de coordenadas não depende da orientação do dispositivo.



SensorEventListener

- Interface para *callback* de *SensorEvent*.
  - *void onAccuracyChanged(Sensor sensor, int accuracy)*.
    - Chamado quando a precisão do sensor muda.
  - *void onSensorChanged(SensorEvent event)*.
    - Chamado quando os valores do sensor mudam.

Registro de eventos de Sensor

- Use *SensorManager* para registrar ou cancelar registro de ouvintes de *SensorEvent*.

```
public boolean registerListener(
    SensorEventListener listener, Sensor sensor,
    int rate);
```

  - Registra um *SensorEventListener* para um sensor.

```
public void unregisterListener(
    SensorEventListener listener, Sensor sensor)
```

  - Cancela o registro de um *listener* junto aos sensores em que está registrado.

Exemplo de acelerômetro

```
public class SensorShowValuesActivity extends Activity
    implements SensorEventListener {
  ...

  public void onCreate(Bundle savedInstanceState) {
    ...
    mSensorManager =
      (SensorManager)getSystemService(SENSOR_SERVICE);
    mAccelerometer =
      mSensorManager.getDefaultSensor(Sensor.
          TYPE_ACCELEROMETER);
  }

  ...
```

Exemplo de acelerômetro (continuação)

```
  ...

  protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this,
        mAccelerometer, SensorManager.
        SENSOR_DELAY_NORMAL);
  }

  protected void onPause() {
    mSensorManager.unregisterListener(this);
    super.onPause();
  }
```

Exemplo de acelerômetro (continuação)

```
...

public void onSensorChanged(SensorEvent event) {
  if (event.sensor.getType() == Sensor.
     TYPE_ACCELEROMETER) {
    long actualTime = System.currentTimeMillis();
    if (actualTime - mLastUpdate > 500) {
      mLastUpdate = actualTime;
      float x = event.values[0], y = event.values[1],
          z = event.values[2];
      xView.setText(String.valueOf(x));
      yView.setText(String.valueOf(y));
      zView.setText(String.valueOf(z));
      ...
    }
  }
}
```

Filtragem de valores

- Quando o dispositivo está deitado sobre uma superfície plana, o acelerômetro informa (idealmente) os seguintes valores:
  - $x \approx 0\,\mathrm{m/s^2}$;
  - $y \approx 0\,\mathrm{m/s^2}$;
  - $z \approx 9.81\,\mathrm{m/s^2}$;
- Esses valores podem variar por causa de movimentos naturais, superfícies não planas, ruído, etc.
- Reparar que os valores retornados são acelerações (no sentido dos eixos) em $\mathrm{m/s^2}$.

Filtragem de valores (continuação)

- Dois tipos comuns de filtros:
  - Filtros passa-baixa:
    - Enfatizam componentes de forças constantes.
    - Não enfatizam mudanças de forças transientes.
    - Exemplo de aplicação: nível de bolha.
  - Filtros passa-alta:
    - Não enfatizam componentes de forças constantes.
    - Enfatizam mudanças de forças transientes.
    - Exemplo de aplicação: controles de jogos.

Filtragem de valores (continuação)

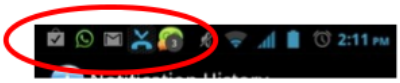
```
mAlpha = 0.9f;

// simple low-pass filter
float lowPass(float current, float filtered) {
  return mAlpha * current + (1.0f - mAlpha) *
      filtered;
}

// simple high-pass filter
float highPass(float current, float last, float
    filtered) {
  return mAlpha * (filtered + current - last);
}
```

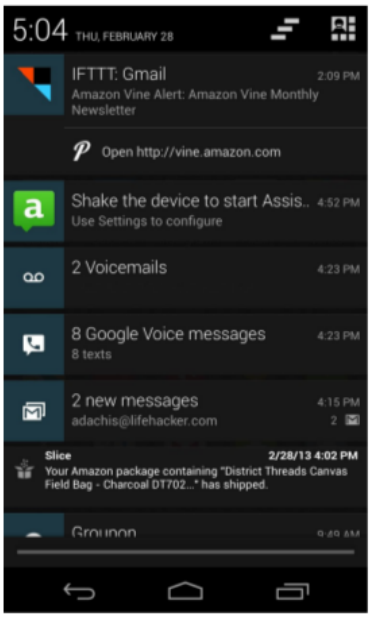
Filtragem de valores (continuação)

```
x = event.values[0];
y = event.values[1];
z = event.values[2];

mLowPassX = lowPass(x, mLowPassX);
mLowPassY = lowPass(y, mLowPassY);
mLowPassZ = lowPass(z, mLowPassZ);

mHighPassX = highPass(x, mLastX, mHighPassX);
mHighPassY = highPass(y, mLastY, mHighPassY);
mHighPassZ = highPass(z, mLastZ, mHighPassZ);

mLastX = x;
mLastY = y;
mLastZ = z;
```

Conteúdo

Notificações





- *Notificação*: mensagem exibida para o usuário fora da área de qualquer aplicação em uma *área de Notificação* (superior).
  - Utilizada para indicar eventos de sistema, *status* de serviços, etc.
- Notificações podem ter:
  - Ícones (pequenos, grandes).
  - Título.
  - Descrição detalhada.
  - Uma ou mais ações associadas, executadas quando a notificação for clicada.

Criação de notificações

- Cria-se uma notificação utilizando-se um *NotificationBuilder*.
- O *NotificationManager* é usado para enviar a notificação.

```
Notification.Builder builder = new Notification.
    Builder(this)
      .setContentTitle("title")
      .setContentText("text")
      .setAutoCancel(true)
      .setSmallIcon(R.drawable.icon);
Notification notification = builder.build();

NotificationManager manager = (NotificationManager)
     getSystemService(Context.NOTIFICATION_SERVICE);
manager.notify(ID, notification);
```

Propriedades de notificações

setAutoCancel(boolean)	Esconder ao clicar?
setColor(argb)	Cor de fundo
setContentIntent(Intent)	*Intent* da ação a rodar quando clicar
setContentText("s")	Descrição detalhada
setContentTitle("s")	Texto de cabeçalho
setGroup("s")	Agrupar notificações semelhantes
setLargeIcon(Bitmap)	Imagem para ícone (grande)
setLights(argb, onMS, offMS)	Luzes piscantes
setNumber(n)	Número à direita da notificação
setSmallIcon(id)	Imagem para ícone (pequeno)
setSound(Uri)	Som a tocar
setTicker("s")	Texto a rolar na barra superior
setVisibility(vis)	Mostrar notificação?
setWhen(ms)	*Timestamp* da notificação

Notificação com ação

- Normalmente, quando o usuário clica em uma notificação, uma ação deve ocorrer (direcionar o usuário para uma aplicação ou atividade, etc.)
  - Para obter isso, use um *intent* dentro de sua notificação.
  - Deve ser empacotado dentro de um objeto de "*intent* pendente".

```
Notification.Builder builder = ...;
Intent intent = new Intent(this, ActivityClassName.
    class);
intent.putExtra("key1", "value1");
...
PendingIntent pending =
    PendingIntent.getActivity(this, 0, intent, 0);
builder.setContentIntent(pending);
Notification notification = builder.build();
...
```