Introdução à Programação Android

Prof. Samir Bonho

Aula 7

Florianópolis, 09 de Abril de 2014

Sumário

- ✓ Sensores
- ✓ Fontes:
  - ✓ Android Developers. Disponível em http://developer.android.com/

Sensores

Android: Sensores

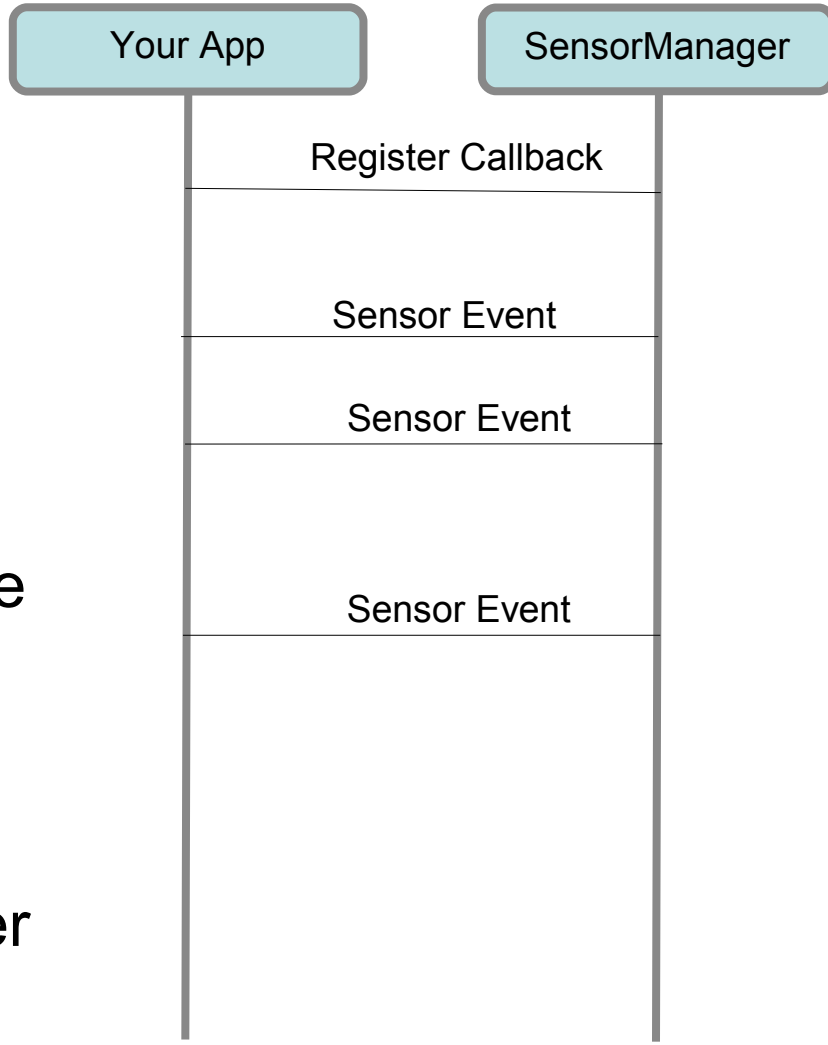
- MIC
- Camera
- Temperatura
- Localização (GPS ou Network)
- Campo magnético (orientação)
- Acelerômtro
- Proximidade
- Pressão
- Luz

Async Callbacks

- Os sensores são controlados por serviços externos e apenas enviam eventos quando se houver demanda específica.

- Uma applicação deve registrar um callback para receber a notificação de eventos nos sensores

- Cada sensor está relacionado com um Listener cujos métodos devem ser implementados e tratados pelo callback.



Utilizando o serviço correto

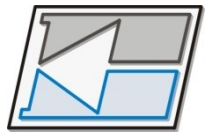
- Os sensores são gerenciados por uma variedade de classes *XXXXManager*:
  - LocationManager (GPS)
  - SensorManager (acelerômetro, giroscópio, proximidade, etc)
- Cada Activity deve chamar a função getSystemService() para obter uma instância do *manager*

```
public class MyActivity … {

  private SensorManager sensorManager_;

  public void onCreate(){
     …

    sensorManager_ = (SensorManager) getSystemService(SENSOR_SERVICE);
  }
}
```



Recebendo atualizações dos sensores

- O *SensorManager* gerencia os eventos de:
  - Acelerômetro, Temperatura, Luz, Giroscópio

- Para receber atualizações com as medidas dos sensores é necessário implementar um interface *SensorEventListener*

- Uma vez a classe *SensorManager* instanciada, é preciso obter uma referência ao sensor de interesse

- O método *registerListener* determina qual sensor que será lido e a velocidade de atualização dos valores

Recebendo atualizações dos sensores

```
public class MyActivity … implements SensorEventListener{
  private Sensor accelerometer_;
  private SensorManager sensorManager_;

  public void connectToAccelerometer() {
            sensorManager_ = (SensorManager)getSystemService(SENSOR_MANAGER);
            accelerometer_ = sensorManager_

.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
            sensorManager_.registerListener(this, accelerometer_,
                                                SensorManager.SENSOR_DELAY_NORMAL);
}
```

int	SENSOR_DELAY_FASTEST	get sensor data as fast as possible
int	SENSOR_DELAY_GAME	rate suitable for games
int	SENSOR_DELAY_NORMAL	rate (default) suitable for screen orientation changes
int	SENSOR_DELAY_UI	rate suitable for the user interface

Interface SensorEventListener

- Há apenas uma interface de escuta para diferentes tipos de sensores.

- Para receber dados de múltiplos sensores, é necessário verificar qual sensor está enviando o evento e tratar seus dados em diferentes trechos de código.

```
public class MyActivity … implements SensorEventListener{

        // Called when a registered sensor changes value
            @Override
            public void onSensorChanged(SensorEvent sensorEvent) {
                    if (sensorEvent.sensor.getType() ==
Sensor.TYPE_ACCELEROMETER) {
                                float xaccel = sensorEvent.values[0];
                                float yaccel = sensorEvent.values[1];
                                float zaccel = sensorEvent.values[2];
                    }
            }
            // Called when a registered sensor's accuracy changes
            @Override
            public void onAccuracyChanged(Sensor arg0, int arg1) {
                    // TODO Auto-generated method stub


            }
}
```

Interface SensorEventListener

```
public class MyActivity … {

    private class AccelListener implements SensorEventListener {
            public void onSensorChanged(SensorEvent sensorEvent) {
                …
            }
            public void onAccuracyChanged(Sensor arg0, int arg1) {}
    }

    private class LightListener implements SensorListener {
            public void onSensorChanged(SensorEvent sensorEvent) {
                …
            }
            public void onAccuracyChanged(Sensor arg0, int arg1) {}
    }
    …
```

Interface SensorEventListener

```
…
private SensorListener accelListener_ = new AccelListener();
private SensorListener lightListener_ = new LightListener();


…
public void onResume(){
   …
 sensorManager_.registerListener(accelListener, accelerometer,
                                        SensorManager.SENSOR_DELAY_GAME);
 sensorManager_.registerListener(lightListener, lightsensor,
                                        SensorManager.SENSOR_DELAY_NORMAL);

}
public void onPause(){
  sensorManager_.unregisterListener(accelListener_);
    sensorManager_.unregisterListener(lightListener_);
}
```

Exercícios