

Nuevos Paradigmas de interacción

Practica 3: Android

Enrique García González
<https://github.com/nRaiker/Practica3-NPI>

Introducción

En esta práctica el problema más importante a tratar es el del uso de los sensores que traen los dispositivos móviles Android para la realización de una serie de aplicaciones.

En primer lugar, para poder utilizar los sensores del dispositivo debemos hacer que las clases que usen los sensores implementen la interfaz `SensorEventListener`. Para poder escuchar los sensores, debemos declarar un objeto del tipo `SensorManager`, que nos permitirá inicializar objetos del tipo sensor de forma correcta.

```
public class SensorActivity extends Activity, implements SensorEventListener {
    private final SensorManager mSensorManager;
    private final Sensor mAccelerometer;

    public SensorActivity() {
        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        mAccelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    }
}
```

En la clase `onResume` del activity debemos llamar a la función `registerListener` para que se inicie la recogida de datos de los sensores indicados, y en el método `onPause` es recomendable llamar a la función `unregisterListener` para detener la actividad de los sensores y ahorrar batería.

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

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

Por último, para poder usar la interfaz debemos implementar los métodos `onAccuracyChanged` y `onSensorChanged`. Estos métodos pueden no hacer nada, pero si queremos que nuestra aplicación reaccione a los cambios de los sensores deberemos implementar estas cosas en el método `onSensorChanged`.

```
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}

public void onSensorChanged(SensorEvent event) {
}
}
```

Para más información sobre los distintos tipos de sensores que podemos utilizar y los métodos que tienen se puede consultar el siguiente enlace:

<http://developer.android.com/intl/es/reference/android/hardware/Sensor.html>

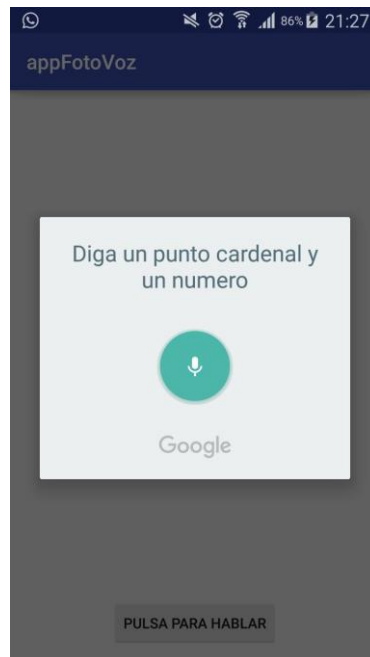
Aplicación BrujulaVoz

El funcionamiento de la aplicación BrujulaVoz consiste en recibir mediante el micrófono una frase que contenga un punto cardinal y un número entero. Una vez hecho esto se debe reconocer la cadena y mostrar en pantalla una brújula apuntando hacia el punto cardinal y cuando tengamos el teléfono apuntando hacia ese punto cardinal debe de mostrar algún tipo de señal.

Para poder a través del teléfono reconocer una frase y almacenar su contenido escrito en el teléfono debemos utilizar el reconocedor de voz proporcionado por google. Para poder hacer esto debemos tener activado el servicio desde un principio en nuestro teléfono, y después dentro de la aplicación llamar al servicio de reconocimiento de voz. La forma hacerlo sería similar a la siguiente, donde elegimos el método de entrada del lenguaje con la primera llamada a `putExtra`, y le pasamos también el texto que queremos que muestre el servicio de reconocimiento de voz cuando se llama:

```
private void startVoiceRecognitionActivity() {  
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
  
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,  
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);  
  
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Diga un punto cardinal y un numero");  
  
    startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);  
}
```

Cuando ejecutamos esta función en la aplicación obtenemos lo siguiente:



Para la implementación de la brújula he utilizado los sensores del acelerómetro y del magnetómetro. Con el magnetómetro podemos saber en qué dirección se encuentra el campo magnético de la tierra y con el acelerómetro obtenemos la orientación y la inclinación del teléfono, lo que nos ayuda a saber hacia dónde debe apuntar la brújula.

Para la implementación de la brújula he utilizado parte del código de Aritz Gamboa que se encuentra en la siguiente página: <http://agamboadev.esy.es/como-crear-un-brujula-en-android/>

A este código hay que añadirle las modificaciones necesarias para que pueda detectar cualquiera de los puntos cardinales y no solo el norte, y la capacidad de cambiar el color de la brújula cuando el teléfono está apuntando a la posición indicada.

Un ejemplo de uso de la aplicación es: el siguiente:

