

Nuevos Paradigmas de Interacción

Rubén Peralta Díaz

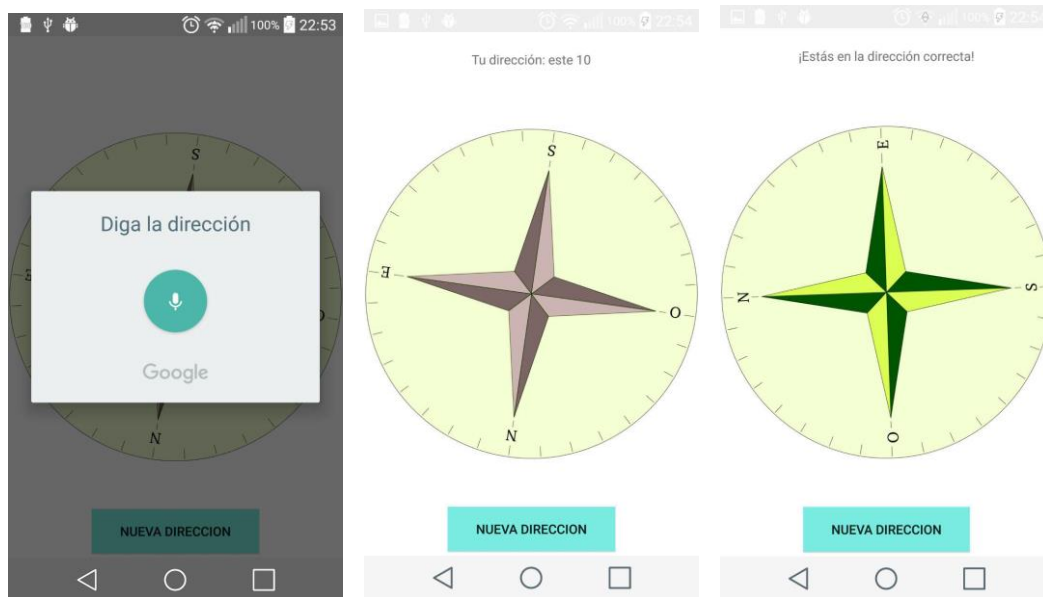
Ahisahar Pretel Rodríguez

Tutorial de Android – Brújula y Reconocimiento de Voz

En este tutorial vamos a mostrar los aspectos más importantes a la hora de realizar una aplicación de brújula con reconocimiento de voz para Android.

El funcionamiento de la aplicación es muy sencillo. El usuario indicará a esta la dirección en la que se quiere orientar pulsando en un botón e indicando la dirección oralmente, junto con el porcentaje de error con el que está dispuesto el usuario a orientarse. Una vez indicada la dirección, con la ayuda de la brújula el usuario podrá orientarse fácilmente, y cuando esté mirando hacia la dirección correcta, la aplicación se lo indicará volviendo la brújula verde y con un mensaje en la pantalla.

Aquí podemos ver un ejemplo de uso:



Para la realización de la aplicación, primero hemos creado un nuevo proyecto con una actividad en blanco y hemos personalizado el aspecto de la aplicación en el archivo correspondiente al Layout.

En este caso, la modificación ha sido muy sencilla, ya que lo único que hemos hecho es quitar los elementos que no necesitábamos del layout por defecto y añadir los objetos necesarios para visualizar la imagen de la brújula, el botón para indicar la dirección y el campo de texto para los mensajes.

```

<TextView
    android:id="@+id/tvHeading"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:layout_alignParentTop="false"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp" />

<ImageView
    android:id="@+id/rosadelosvientos"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:src="@drawable/rosadelosvientos2"
    android:contentDescription="Imagen de una brujula"
    android:layout_centerVertical="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="100dp" />

<Button
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:text="Nueva direccion"
    android:id="@+id/button"
    android:background="#7bebe2"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="5dp" />

```

A continuación, pasamos a modificar el código que controla la lógica de la aplicación, la actividad principal. Lo primero que debemos hacer es añadir en la definición de la clase *'implements SensorEventListener'*, ya que necesitaremos hacer uso de los sensores del dispositivo.

El siguiente punto importante en la implementación de la aplicación es la llamada al reconocimiento de voz. Para ello debemos utilizar el método *'setOnClickListener'* en el botón que da comienzo al reconocimiento de voz y sobrescribir el método *'onClick'* en el que crearemos el *'Intent'* con el que llamaremos a la síntesis de voz de google:

```

this.botonHablar = (Button) this.findViewById(R.id.button);
this.botonHablar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        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 la dirección");
        intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 3);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, new Locale("spa", "ESP"));
        startActivityForResult(intent, ASR_CODE);
    }
});

```

El resultado del reconocimiento de habla lo procesaremos en el método *'fromActivityResult'*, en el que primero comprobamos si el código de la solicitud es el que enviamos nosotros (en ese caso estamos procesando el resultado de nuestra llamada), y después utilizamos los datos, si es que ha habido algún resultado.

```
@SuppressWarnings("InlinedApi")
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == ASR_CODE) {
        if (resultCode == RESULT_OK) {
            if (data != null) {
                ArrayList<String> nBestList = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                String[] palabras = nBestList.get(0).split(" ");
                if (palabras.length == 2 && formatoCorrecto(palabras[0]) && numerico(palabras[1])) {
```

El último método importante que debemos implementar es el que recibe la información de los sensores que utilizaremos para orientar la brújula. Este método es *'onSensorChanged'*.

En este método obtenemos el ángulo de la orientación del móvil con respecto al Norte, y podemos entonces hacer los cálculos para ver si estamos bien orientados según la dirección que el usuario indicó, y también girar la imagen de la brújula mediante un objeto de la clase *RotateAnimation*.

```
public void onSensorChanged(SensorEvent event) {

    // Obtiene el ángulo
    float degree = Math.round(event.values[0]);

    // create a rotation animation (reverse turn degree degrees)
    RotateAnimation ral = new RotateAnimation(
        angulo,
        -degree,
        Animation.RELATIVE_TO_SELF, 0.5f,
        Animation.RELATIVE_TO_SELF,
        0.5f);

    // how long the animation will take place
    ral.setDuration(210);
    // set the animation after the end of the reservation status
    ral.setFillAfter(true);
    // Start the animation
    imagen.startAnimation(ral);
    angulo = -degree;

    if (en_direccion == true) {
        if (degree < dest + (360 * tol / 2) / 100 && degree > dest - (360 * tol / 2) / 100) {
            imagen.setImageResource(R.drawable.rosadelosvientos);
            tvDir.setText("¡Estás en la dirección correcta!");
        } else {
            imagen.setImageResource(R.drawable.rosadelosvientos2);
            tvDir.setText("Tu dirección: ".concat(direccion).concat(" ").concat(Integer.toString(tol)));
        }
    }
}
```

Referencias:

<https://nuevos-paradigmas-de-interaccion.wikispaces.com/file/detail/Tutorial%20Brujula.pdf>

<http://developer.android.com/intl/es/reference/android/view/animation/RotateAnimation.html>