

Ejercicios de geolocalización y mapas

Índice

1 Geolocalización.....	2
2 Geocoder (*).....	2
3 Mapas.....	3
4 Marcadores (*).....	3
5 Reconocimiento del habla (*).....	4

1. Geolocalización

Haremos que la aplicación `Geolocalizacion` nos localice geográficamente utilizando GPS, y nos muestre nuestras coordenadas actuales en pantalla.

Se pide:

- a) Al crear la actividad, mostrar en los campos `txtLat` y `txtLon` la latitud y la longitud de la última lectura registrada hasta el momento.
- b) Pedir actualizaciones de la lectura del GPS cada 5 segundos, siempre que nos hayamos movido al menos 10 metros. Cada vez que se reciba una lectura, actualizaremos los campos anteriores.

Para poder probar esto en el emulador deberemos indicarle manualmente al emulador las coordenadas en las que queremos que se localice. Esto lo podemos hacer de dos formas: mediante línea de comando o mediante la aplicación DDMS. Vamos a verlas a continuación.

Para comunicar las coordenadas al emulador mediante línea de comando deberemos conectarnos a él mediante `telnet`. Por ejemplo, si nuestro emulador está funcionando en el puerto 5554, haremos un `telnet` a `localhost` y a dicho puerto:

```
telnet localhost 5554
```

Una vez dentro de la línea de comando del emulador, invocaremos el comando `geo` para suministrarle las coordenadas. Por ejemplo, las siguientes coordenadas corresponden a la Universidad de Alicante:

```
geo fix -0.51515 38.3852333
```

Si no queremos tener que ir a línea de comando, podemos utilizar la aplicación DDMS a la que se puede acceder de forma independiente o desde dentro de Eclipse. Dado que estamos ejecutando el emulador desde Eclipse, deberemos lanzar DDMS también dentro de este entorno. Para ello deberemos mostrar la vista *Emulator Control*. En ella veremos unos cuadros de texto y un botón con los que enviar las coordenadas al emulador, siempre que esté en funcionamiento.

Advertencia

Debido a un *bug* del SDK de Android, en determinadas versiones el DDMS no envía correctamente las coordenadas al emulador si nuestro *locale* no está configurado con idioma inglés. Para solucionar esto de forma sencilla, podemos editar el fichero `eclipse.ini` y añadir dentro de él la opción `-Duser.language=en`. Si no hacemos esto, el emulador recibirá siempre las coordenadas 0, 0.

2. Geocoder (*)

Continuaremos trabajando con el proyecto anterior, esta vez para hacer que al obtener una nueva lectura nos muestre también nuestra dirección en forma de texto utilizando el *geocoder*.

Implementaremos la obtención de la dirección en el método `obtenerDireccion`, que deberá llamar al *geocoder* proporcionando las coordenadas obtenidas en la última lectura. Puedes utilizar para ello los campos `latitud` y `longitud` de la clase. Cada vez que se produzca una nueva lectura del GPS guardaremos las coordenadas en dichos campos, y ejecutaremos la tarea `GeocodeTask` para que obtenga en segundo plano la dirección a la que corresponden dichas coordenadas y la muestre en pantalla.

Para utilizar el *geocoder*, deberemos utilizar un emulador que incorpore las APIs de Google (para así poder acceder a la API de mapas). Además, dado que necesitará conectarse a Internet para obtener las direcciones, deberemos solicitar el permiso `INTERNET`.

Advertencia

En el emulador de la plataforma Android 2.2 no funciona correctamente el *geocoder*. Funcionará correctamente si utilizamos, por ejemplo, un emulador con Google APIs de nivel 3 (versión 1.5 de la plataforma).

3. Mapas

Vamos a trabajar con la aplicación `Mapas` que integra los mapas de Google. Se pide:

- a) Para que los mapas se muestren correctamente lo primero que necesitaremos es obtener una clave de la API de mapas y especificarla dentro del `layout main.xml` donde tenemos definido el `MapView`. La clave se deberá obtener vinculada al certificado *debug* de nuestra máquina de desarrollo. Comprueba que tras hacer esto el mapa se muestra al ejecutar la aplicación.
- b) Configura el mapa al crear la actividad `MapasActivity` (en la función `inicializaMapa`) para que muestre controles de zoom e imagen de satélite.
- c) Establece el nivel de zoom 17, y centra el mapa en la Universidad de Alicante (38.3852333, -0.51515).

4. Marcadores (*)

Vamos ahora a añadir marcadores al mapa anterior. Se pide:

- a) Añade un marcador con nuestra posición actual. Haz que nuestra posición deje de actualizarse cuando la actividad pase a estar pausada, y que vuelva a actualizarse cuando

se reanude.

b) Haz que el mapa se centre en nuestra posición cuando ésta sea obtenida. Deberá moverse a nuestra posición mediante una animación.

c) Añade marcadores al mapa para las localizaciones definidas en la lista `localizaciones`, utilizando un objeto `ItemizedOverlay`. Como *drawable* para los marcadores utilizaremos `R.drawable.marker`.

d) Cambia el `ItemizedOverlay` anterior por un `BalloonItemizedOverlay`, para mostrar globos informativos en cada marcador.

e) Haz que los marcadores de los restaurantes de muestren de color verde (`R.drawable.marker2`), y los de transportes en rojo (marcador por defecto).

5. Reconocimiento del habla (*)

Probar la aplicación `Habla`. Esta aplicación tiene un campo de texto y un botón. Al pulsar sobre el botón se lanzará el módulo de reconocimiento del habla, y una vez finalizado mostraremos lo que se haya reconocido en el campo de texto.

Nota

Sólo podremos probar este ejemplo si contamos con un dispositivo real, ya que el emulador no soporta el reconocimiento del habla.

