

25. Menús

Los menús muestran gráficamente opciones que, al ser seleccionadas, arrancan la tarea correspondiente. Básicamente, existen dos tipos de menús en Android:

- El menú de **opciones** de la actividad actual que se activa cuando pulsas el botón `MENU`. Las opciones se pueden mostrar como texto, casillas de selección (check boxes) o botones de selección (radio buttons).
- El menú de **contexto** de una vista concreta, que se muestra cuando mantienes la pulsación sobre la vista. Las opciones se pueden mostrar como casillas de selección o botones de selección. Este menú también permite añadir teclas de atajo (shortcutkeys) y submenús.

Un submenú se muestra como una ventana flotante que aparece cuando se pulsa una cierta opción del menú que lo lleva aparejado.

Para mostrar el menú de opciones de una actividad determinada hay que implementar dos métodos de la actividad:

- `onCreateOptionsMenu()`: este método se ejecuta cuando pulsas el botón `MENU`. Aquí debes suministrar código de interfaz de usuario.
- `onOptionsItemSelected()`: este método se ejecuta cuando se selecciona un ítem del menú creado en `onCreateOptionsMenu()`. El código del método debe llevar a cabo la tarea correspondiente al ítem seleccionado.

El proyecto de esta unidad, CCC25, añade un menú de opciones a nuestro juego. Recordemos que nuestro juego cuenta ya con cuatro clases: `Game`, `Initial`, `MainActivity` y `Music`, como puedes comprobar en el proyecto CCC22. Vamos a añadir la clase `About` y a modificar `MainActivity` para incluir el menú de opciones que va a permitirnos enviar mensajes como vimos en la unidad 23 y mostrar información sobre nuestra aplicación:

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="es.uam.eps.android.CCC25"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".Initial"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

        </activity>
        <activity
            android:name=".MainActivity"
            android:label="" >
            <intent-filter>
                <action android:name="es.uam.eps.android.CCC25.MAINACTIVITY" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name=".About"
            android:theme="@android:style/Theme.Holo.Dialog"
            android:label="" >
            <intent-filter>
                <action android:name="es.uam.eps.android.CCC25.ABOUT" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Una forma sencilla de especificar la interfaz de usuario del menú es crear un fichero XML y colocarlo en la carpeta /res/menu (añádela a res pulsando con el botón derecho del ratón sobre la carpeta res):

/res/menu/ccc_menu.xml

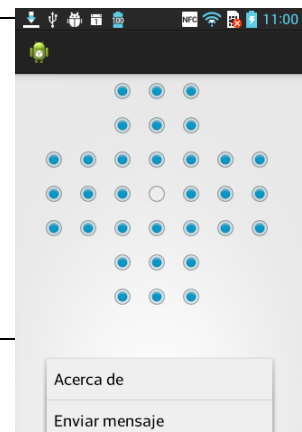
```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/menuAbout"
        android:icon="@android:drawable/ic_menu_info_details"
        android:title="@string/aboutText"/>
    <item
        android:id="@+id/sendMessage"
        android:icon="@android:drawable/ic_dialog_email"
        android:title="@string/sendMessageText"/>

</menu>

```



El elemento `menu` corresponde al conjunto del menú y cada elemento `item` especifica un ítem del menú con las siguientes propiedades:

- título (`android:title`)
- identificador (`android:id`)
- icono (`android:icon`)

El título recibe un valor mediante un recurso especificado en el fichero `strings.xml` del que, mas adelante, haremos una versión en inglés para internacionalizar el juego:

/res/values/strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">CCC25</string>
    <string name="action_settings">Ajustes</string>
    <string name="gameOverTitle">Fin del juego</string>
    <string name="initialImage">Imagen inicial del juego</string>
    <string name="aboutText">Acerca de</string>
    <string name="sendMessageText">Enviar mensaje</string>

</resources>

```

```

<string name="searchText">Buscar</string>
<string name="searchByNameText">Buscar por nombre</string>
<string name="searchByBirthDateText">Buscar por fecha de nacimiento</string>
<string name="searchByLastNameText">Buscar por apellido</string>
<string name="aboutMessage">Cha cha cha es un juego solitario
    en el que hemos de comer una a una las fichas
    saltando por encima de ellas, solo en vertical u
    horizontal, hacia la casilla adyacente que se
    encuentre vacía. El objetivo es dejar
    la última ficha en el centro del tablero, el
    lugar que comenzó libre.</string>
</resources>

```

La interfaz gráfica del menú se infla en el método `onCreateOptionsMenu()` de la clase `MainActivity`:

```
/src/MainActivity.java
```

```

...
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.ccc_menu, menu);
    return true;
} ...

```

La tarea ejecutada al seleccionar un ítem determinado se codifica en el método `onOptionsItemSelected()`:

```
/src/MainActivity.java
```

```

...
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menuAbout:
            startActivity(new Intent(this, About.class));
            return true;
        case R.id.sendMessage:
            Intent intent = new Intent(android.content.Intent.ACTION_SEND);
            intent.setType("text/plain");
            intent.putExtra(Intent.EXTRA_SUBJECT, "CHA CHA CHA");
            intent.putExtra(Intent.EXTRA_TEXT, "Hola ..., he llegado a ... puntos en cha
cha cha ...");
            startActivity(intent);
            return true;
    }

    return super.onOptionsItemSelected(item);
} ...

```

Por ejemplo, si el usuario pulsa el ítem About, se ejecuta la actividad `About`:

```
startActivity(new Intent(this, About.class));
```

El método `onCreate()` de `About` infla el interfaz especificado en `about.xml`:

```
/src/About.java
```

```

package es.uam.eps.android.CCC25;

import android.app.Activity;
import android.os.Bundle;

public class About extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.about);
    }
}

```

El fichero `about.xml` es el siguiente:

`/res/layout/about.xml`

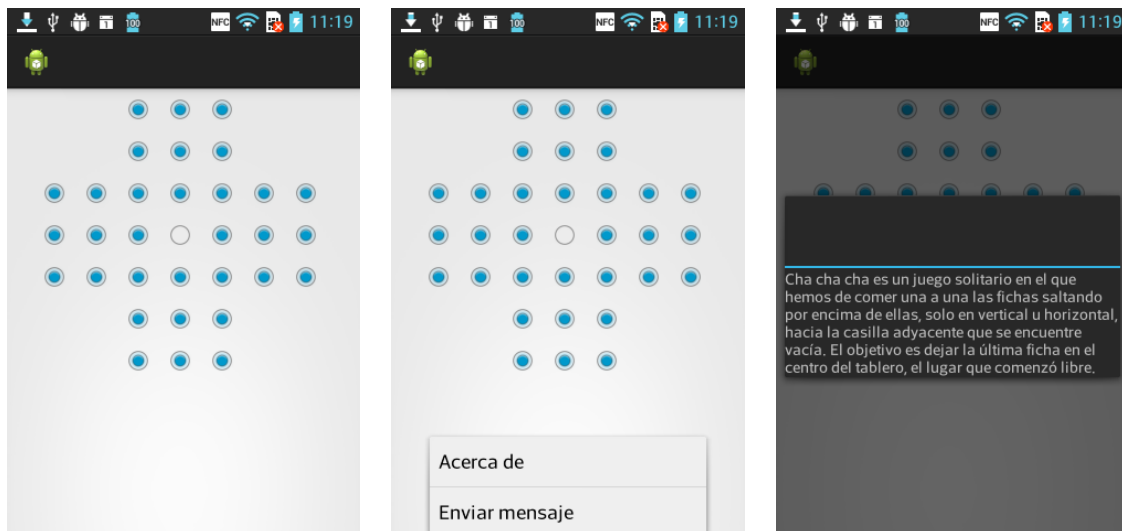
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/aboutMessage" />

</LinearLayout>
```

A continuación se muestra un ejemplo de sucesión de acciones hasta ejecutar About:

- Durante la ejecución de `MainActivity` pulsamos la tecla del menú de opciones.
- Aparece el menú de opciones.
- Pulsamos el ítem Acerca de.
- Aparece una ventana ligera mostrando las reglas del juego.



Como ves, la actividad `About` no tapa por completo al tablero sino que se muestra como si fuera una ventana de diálogo gracias al tema especificado en el fichero de manifiesto:

```
<activity
    android:name=".About"
    android:theme="@android:style/Theme.Holo.Dialog"
    android:label="" >
    <intent-filter>
        <action android:name="es.uam.eps.android.ccc25.ABOUT" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

25.1 Submenús

Para programar un submenú basta anidar un elemento `menu` dentro de uno de los ítems del fichero de diseño. Por ejemplo, añadamos un submenú a un nuevo ítem del menú anterior que llamaremos Search. El nuevo fichero de diseño es el siguiente:

/res/menu/ccc_menu.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:title="@string/aboutText"
        android:id="@+id/menuAbout"
        android:icon="@android:drawable/ic_menu_info_details"/>
    <item
        android:title="@string/sendMessageText"
        android:id="@+id/sendMessage"
        android:icon="@android:drawable/ic_dialog_email"/>
    <item
        android:title="@string/searchText"
        android:id="@+id/menu_search"
        android:icon="@android:drawable/ic_menu_search">
        <menu>
            <group android:checkableBehavior="single">
                <item
                    android:id="@+id/search_by_name"
                    android:title="@string/searchByNameText"
                    android:checked="true"/>
                <item
                    android:id="@+id/search_by_birth_date"
                    android:title="@string/searchByBirthDateText"/>
                <item
                    android:id="@+id/search_by_last_name"
                    android:title="@string/searchByLastNameText"/>
            </group>
        </menu>
    </item>
</menu>
```

El elemento `<group>` sirve para agrupar opciones y aplicarles ciertos atributos colectivamente. Por ejemplo, `android:checkableBehavior` sirve para especificar si los ítems deben aparecer como ítems sencillos, casillas o botones. Sus valores pueden ser:

- `single`: botones donde solo un ítem se puede seleccionar
- `all`: casillas donde se pueden seleccionar varios ítems
- `none`: cada ítem se muestra como texto sin casillas ni botones.

Esto es lo que vemos al pulsar la tecla del menú de opciones y luego la opción Buscar:

