Introducción al diseño de interfaces gráficas en Android - Ejercicios

Índice

1. LinearLayout

Crea una aplicación llamada *LinearLayout*. La aplicación contendrá una única actividad, llamada *Principal*, cuya interfaz gráfica estará contruida exclusivamente a partir de layouts de tipo `LinearLayout` y deberá ser lo más parecida posible a la mostrada en la siguiente imagen.

Interfaz gráfica de la aplicación LinearLayout

Nota:

Las líneas han sido creadas por medio de elementos `View` a los que se les ha asignado una altura de *1dip* mediante el atributo `android:layout_height` y un color de fondo *#FFFFFF* mediante el atributo `android:background`.

2. Colores

Creamos ahora una nueva aplicación; en este caso su nombre será *Colores*. La interfaz de su única actividad (también llamada *Principal*) contendrá dos grupos de botones de radio y un checkbox, además de un elemento de tipo `TextView` en la parte superior.

El primer grupo de botones de radio servirá para modificar el color de fondo del elemento `TextView`, mientras que el segundo permitirá modificar el color del texto. Con respecto al checkbox, éste indicará la presencia o no de texto. Con el checkbox activado el texto se mostrará; en caso contrario no se mostrará texto en el `TextView` y tan sólo será visible su color de fondo.

El estado inicial de los elementos de la interfaz gráfica de la actividad *Principal* será el mostrado en la siguiente imagen:

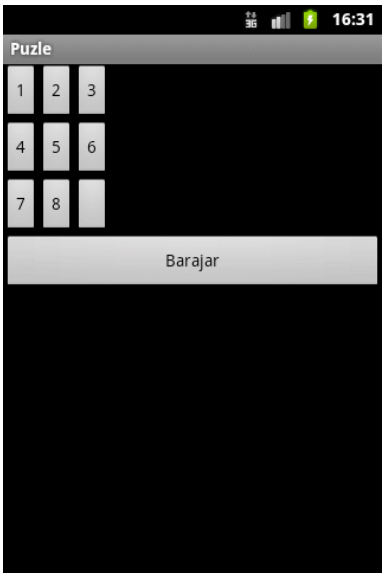
Interfaz gráfica de la aplicación Colores

Nota:

En el código podemos cambiar el color de fondo y de texto de un `TextView` con los métodos `setBackgroundColor` y `setTextColor`. Ambos aceptan como parámetro un entero, que puede ser cualquiera de las constantes estáticas definidas en la clase `Color`.

3. Puzle (*)

En este ejercicio implementaremos una versión en Android de un conocido tipo de puzle. En este caso crearemos una nueva aplicación llamada *Puzle* que como en casos anteriores tan sólo contendrá una actividad, de nombre *Principal*. Su interfaz gráfica, que deberá utilizar al menos un `TableLayout`, tendrá el siguiente aspecto:



Interfaz gráfica de la aplicación Puzle

Como se puede observar, tenemos nueve botones formando una rejilla de 3x3 y numerados del 1 al 8 (uno de los botones no tiene etiqueta). Primero debemos añadir manejadores a los botones de tal forma que si se pulsa un botón adyacente ortogonalmente (es decir, no en diagonal) al botón sin etiqueta, se intercambie la etiqueta entre ambos.

En la parte inferior de la interfaz hay otro botón etiquetado como *Barajar*. Al pulsarlo se deberán realizar cincuenta movimientos aleatorios del puzle para desordenarlo, partiendo de la situación en la que se encontrara el puzle en ese momento, y siguiendo las especificaciones del párrafo anterior: en cada iteración intercambiamos la etiqueta del botón vacío con la de algún botón adyacente ortogonalmente.

4. Ciudades

En este ejercicio practicaremos con los elementos de tipo `Spinner`. La aplicación *Ciudades* contendrá una única actividad de nombre *Principal*. La interfaz de dicha actividad estará compuesta por un `TextView` y dos elementos de tipo `Spinner`. El primero de ellos permitirá escoger entre tres países cualquiera (inicialmente ningún país estará seleccionado). El segundo permitirá escoger una ciudad según el país seleccionado en el anterior. Cada vez que se seleccione un país en el primer `Spinner` deberán cambiar las opciones del segundo, mostrando dos ciudades del país seleccionado.

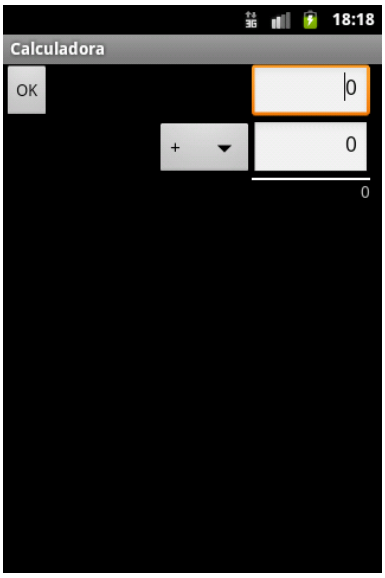
La ciudad seleccionada en el segundo `Spinner` aparecerá en el `TextView` de la parte superior.

Para completar el ejercicio debes seguir los siguientes pasos:

- Añade el `TextView` y los dos `Spinner` al recurso layout de la aplicación, sin olvidar añadir a estos dos últimos su correspondiente atributo `android:prompt` (con los textos "Selecciona país" y "Selecciona ciudad" respectivamente).
- Crea las opciones para los `Spinner` en el archivo *arrays.xml* de los recursos. Elige tú mismo el nombre de los países y de las ciudades. Recuerda que debes crear tres conjuntos diferentes de opciones (tres elementos de tipo `string-array`) para el segundo `Spinner`, ya que las ciudades a escoger cambiarán según el pais escogido en el primero.
- Rellena el primer `Spinner` con sus correspondientes opciones.
- Rellena el segundo `Spinner` con las ciudades correspondintes al primer país. Esto debes hacerlo así porque siempre que inicies la actividad será el primer país el que se encuentre seleccionado.
- Asígnale al TextView como valor inicial el nombre de la primera ciudad, pues será la que se encontrará seleccionada al iniciar la actividad.
- Añade un manejador al `Spinner` de paises para que cada vez que se seleccione una opción se muestren las opciones adecuadas en el `Spinner` de ciudades.
- Añade un manejador al `Spinner` de ciudades para que cada vez que se seleccione una opción se muestre en el `TextView`. Para obtener el texto correspondiente a la opción seleccionada en el `Spinner` puedes utilizar el método `getSelectedItem` del mismo. Una vez hecho esto puedes llamar al método `toString` para obtener la cadena correspondiente.

5. Calculadora sencilla

El objetivo de este ejercicio es implementar una calculadora sencilla. La aplicación *Calculadora* contendrá una única actividad de nombre *Principal*, vuya interfaz gráfica tendrá el siguiente aspecto:



Interfaz gráfica de la aplicación Calculadora

Como se puede observar nuestra calculadora es bastante limitada. Tan solo acepta dos operandos (que se podrán introducir en los dos `EditText`) y cuatro operaciones seleccionables con el `Spinner`: +, -, * y /. En el `TextView` inferior deberá aparecer el resultado de la operación cuando se pulse el botón *Ok*.

A la hora de diseñar la interfaz se ha utilizado un `RelativeLayout`. Los atributos más importantes utilizados han sido: `layout_alignParentRight`, `layout_below`, `align_marginRight`, `android:inputType="number"` para los `EditText` y `android:gravity="right"` para el `TextView` y los `EditText`.