

Drawables, estilos y temas

Índice

| | |
|------------------------------------|---|
| 1 Personalización del aspecto..... | 2 |
| 2 Personalización de botones..... | 2 |
| 3 Animación por fotogramas..... | 2 |
| 4 Niveles..... | 3 |
| 5 Estilos y temas (*)..... | 3 |

1. Personalización del aspecto

Crea una nueva aplicación de Android llamada *Drawables* y haz que muestre un `TextView` que ocupe toda la anchura de la pantalla y que tenga el siguiente aspecto:

- Un marco de color azul con esquinas redondeadas.
- El marco debe tener un grosor de 5 density pixels.
- Se dejará un margen de 10 density pixels entre el marco y el texto que contiene
- Se dejará un margen de 5 density pixels entre los bordes de la pantalla y el marco.
¿Encontramos algún atributo que haga esto en la definición del drawable? ¿Y en el layout?
- Un relleno de gradiente lineal para el fondo, desde gris claro hasta blanco
- El texto será de color negro.

Ayúdate de la documentación oficial de Android para crear el recurso:

<http://developer.android.com/guide/topics/resources/drawable-resource.html#Shape>

2. Personalización de botones

Vamos a añadir un botón a la aplicación anterior debajo del `TextView`, que llenará todo el ancho. Se proporcionan en las plantillas de la sesión tres imágenes, una para cada uno de los estados del botón (*boton_focused.png*, *boton_normal.png* y *boton_pressed.png*). Se pide:

- Dar inicialmente al botón como aspecto la imagen correspondiente al estado sin pulsar.
- Crear un *state list drawable* para que el botón cambie su aspecto para cada estado.
- Convertir las imágenes a nine-patch y utilizar esta nueva versión en la aplicación.

Nos podemos ayudar de la documentación sobre *state list drawable*:

<http://developer.android.com/guide/topics/resources/drawable-resource.html#StateList>

3. Animación por fotogramas

Vamos a añadir a nuestra actividad, debajo del botón, una vista de tipo `ProgressBar` que muestre que se está realizando un progreso de forma indeterminada. Se pide:

- Ejecutar la aplicación y comprobar como aparece un círculo girando continuamente como indicativo de que se está realizando un progreso. Hacer que el indicador de progreso aparezca centrado en la horizontal (para ello podemos usar el atributo `layout_gravity`).
- Vamos ahora a personalizar la animación que figura en el progreso. En las plantillas

de la sesión tenemos cuatro fotogramas de un armadillo caminando. Crea un *animation-list drawable* a partir de dichos fotogramas, que se reproduzca de forma continuada cambiando de fotograma cada 200ms.

- Reemplaza la animación por defecto del `ProgressBar` por la animación que hemos definido nosotros. Pista: se debe especificar como *drawable*, y el progreso continuo indeterminado (sin dar un porcentaje concreto) que queremos mostrar, se conoce en la API de Android como *indeterminate progress*. Busca entre los atributos de `ProgressBar` el que consideres adecuado.

Nos podemos ayudar de la documentación sobre *animation list drawable*:

<http://developer.android.com/guide/topics/resources/animation-resource.html#Frame>

4. Niveles

En este ejercicio vamos a añadir un `SeekBar` a nuestra aplicación *Drawables*, y haremos que se modifique su aspecto según su valor. Seguimos los siguientes pasos:

- Añadir el `SeekBar` debajo del `ProgressBar` del ejercicio anterior, haciendo que ocupe todo el ancho de la pantalla. Comprobar que se muestra correctamente y que al pulsar sobre él podemos mover el cursor. Por defecto, el nivel de la barra va de 0 a 10000.
- Personalizar el `SeekBar` para que el fondo de la barra cambie de color según el nivel seleccionado. Los colores que debe mostrar son:
 - Verde (#FF00FF00): Niveles de 0 a 2500
 - Amarillo (#FFFF00): Niveles de 2501 a 5000
 - Naranja (#FF8800): Niveles de 5001 a 7500
 - Rojo (#FF0000): Niveles de 7501 a 10000

Crea un *Drawable* de tipo rectángulo con esquinas redondeadas para cada estado. Introdúcelos en un *level list drawable*, y establéclo como *progressDrawable* en la barra.

Nos podemos ayudar de la documentación sobre *level list drawable*:

<http://developer.android.com/guide/topics/resources/drawable-resource.html#LevelList>

5. Estilos y temas (*)

En las plantillas de la sesión se te proporciona una aplicación llamada *Estilos* que contiene una única actividad. Su interfaz gráfica está compuesta por dos `TextView` y dos botones. Se han utilizado algunos atributos para modificar el aspecto de estos elementos. Mientras que algunos valores de estos atributos son compartidos por todas o algunas de las vistas, otros son asignados exclusivamente a una.

En concreto:

- Todos los textos de la actividad aparecen de color rojo
- En el caso de los `TextView`, ambos tienen fondo de color blanco y una fuente de tipo *serif*
- Sin embargo, el tamaño del texto del segundo `TextView` es mayor que el del primero
- Con respecto a los botones, ambos comparten el mismo tamaño
- Sin embargo, el texto de un botón está alineado a la izquierda y el del otro a la derecha

Con tal de practicar lo aprendido en la sección de estilos y temas, deberemos definir lo siguiente:

- Un tema para los atributos compartidos por todos los elementos de la interfaz (texto de color rojo). ¿Se puede modificar el color del texto de los botones de esta forma? ¿Qué otro elemento de la interfaz gráfica a pasado a ser de color rojo?
- Un estilo para el primer `TextView` y un estilo heredado de éste para el segundo
- Un estilo para el primer botón y un estilo heredado de éste para el segundo
- Incorpora también los atributos `android:layout_height` y `android:layout_width` como parte de los estilos, de tal forma que en el archivo de layout los dos únicos atributos que tendrá cada vista serán `style` y `android:text`

