

Seminario de Lenguajes Android

Práctica 3

Nota: En los ejercicios que contienen código se recomienda copiarlo desde la siguiente URL:

<https://gist.github.com/gcaseres/4d46cff85068a662ee11f41a494959d8>

1. Modificar el ejercicio 12 de la práctica 2 para que, al girar el dispositivo, se mantenga el número mostrado en pantalla. Utilice los métodos `onSaveInstanceState`/`onRestoreInstanceState`.
2. Implemente la siguiente Activity con cuatro imágenes y un título, usando un `ScrollView`.



3. Modificar el ejercicio anterior para que muestre una única imagen.
Cada vez que se clickea el botón “Siguiente”, se deberá reemplazar la imagen.
Investigar el uso del mensaje `setImageResource` de la clase `ImageView`.



Al girar la pantalla, no debe cambiarse la imagen y el layout debe visualizarse correctamente.

Nota: Puede almacenar las referencias a las imágenes en un arreglo de la siguiente manera:

```
private int[] imagenes = {R.drawable.portada, R.drawable.interior, R.drawable.foto3};
```

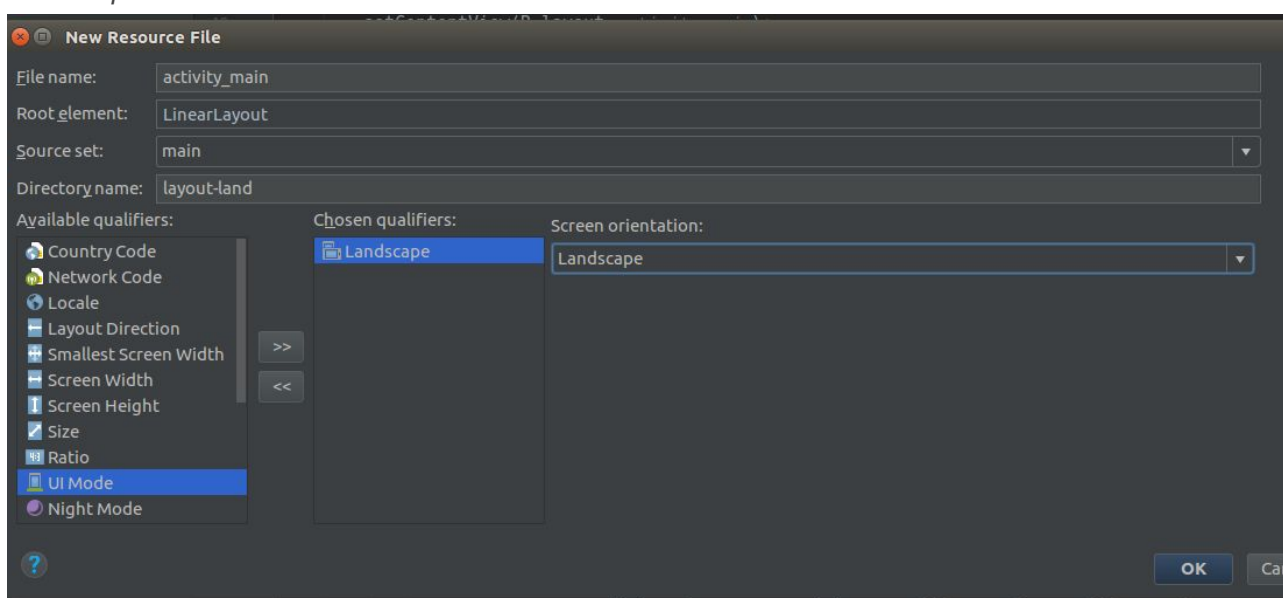
4. Modifique el ejercicio anterior de manera que al girar la pantalla, se modifique el layout de la siguiente forma.



Implementar la grilla de botones usando GridLayout y un XML diferenciado según la orientación.

Nota: para incluir un layout dependiente de la orientación, desde Android Studio, se debe hacer click derecho en el directorio *layout* → *New* → *Layout resource file*.

Deberá tener el mismo nombre de archivo que el layout que usamos para la versión vertical. Luego elegimos “Orientation” en el cuadro de “Available qualifiers”, y luego la orientación *Landscape*:



Por defecto, cuando la aplicación esté en orientación *Portrait*, usará el layout que habíamos definido originalmente, y ahora la orientación *Landscape* usará este nuevo layout.

5. Investigue sobre los recursos de elementos de diseño de Android. (Recursos drawable)
- ¿Para qué sirven?
 - ¿Qué tipos de elementos de diseño están disponibles?
 - ¿Qué tipo de elemento de diseño utilizaría para mostrar una imagen?
 - ¿Qué tipo de elemento de diseño utilizaría para mostrar un rectángulo con colores?
 - ¿Qué tipo de elemento de diseño utilizaría para mostrar los estados “presionado” y “suelto” de un botón?

6. Utilice un recurso drawable para definir el fondo de un TextView que se visualice de la siguiente manera:

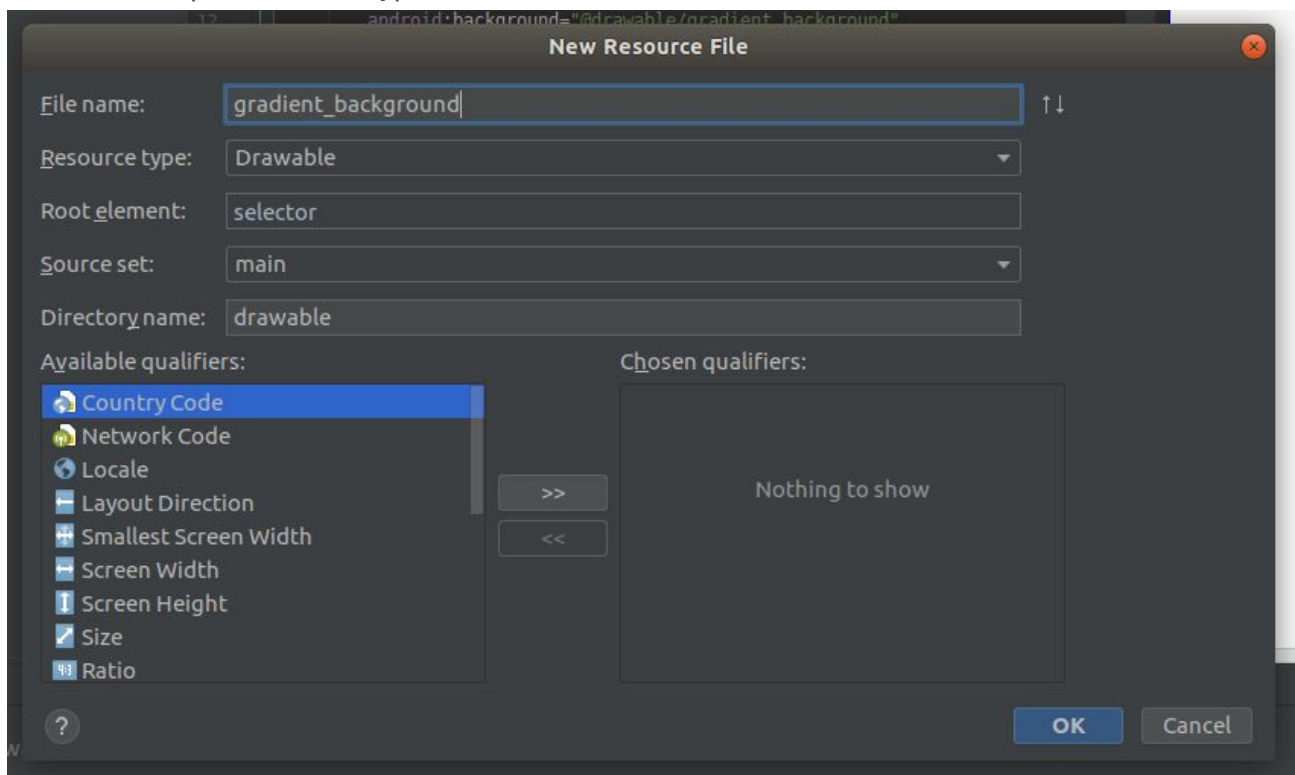


Para ello deberá:

- Generar un recurso drawable denominado a modo de ejemplo: *gradient_background*

Click derecho -> New > Android Resource File

Notar que Resource type debe ser: Drawable



- En el archivo xml generado deberá definir un diseño gradiente vertical lineal entre dos colores. Al mismo tiempo debe definir un borde de 3dp de ancho con un color diferente al del gradiente. Deberá utilizar un Shape. Puede obtener información sobre la sintaxis a utilizar en:

<https://developer.android.com/guide/topics/resources/drawable-resource?hl=es-419#Shape>

- c. Una vez definido el recurso drawable, puede asignárselo a un TextView simplemente estableciendo el atributo android:background de la siguiente manera:

```
<TextView android:background="@drawable/gradient_background"
```

7. Genere un Botón que utilice como background el gradiente definido en el punto anterior. Cuando el usuario presiona el botón, deberá cambiar el gradiente a un color diferente.

Para ello, genere un nuevo recurso similar al del punto anterior pero con colores diferentes para el gradiente con nombre: gradient_background_pressed

A continuación defina un nuevo recurso drawable que defina los dos estados del botón y que haga referencia a los drawables previamente definidos. Deberá utilizar un StateList. Puede obtener información de la sintaxis a utilizar en:

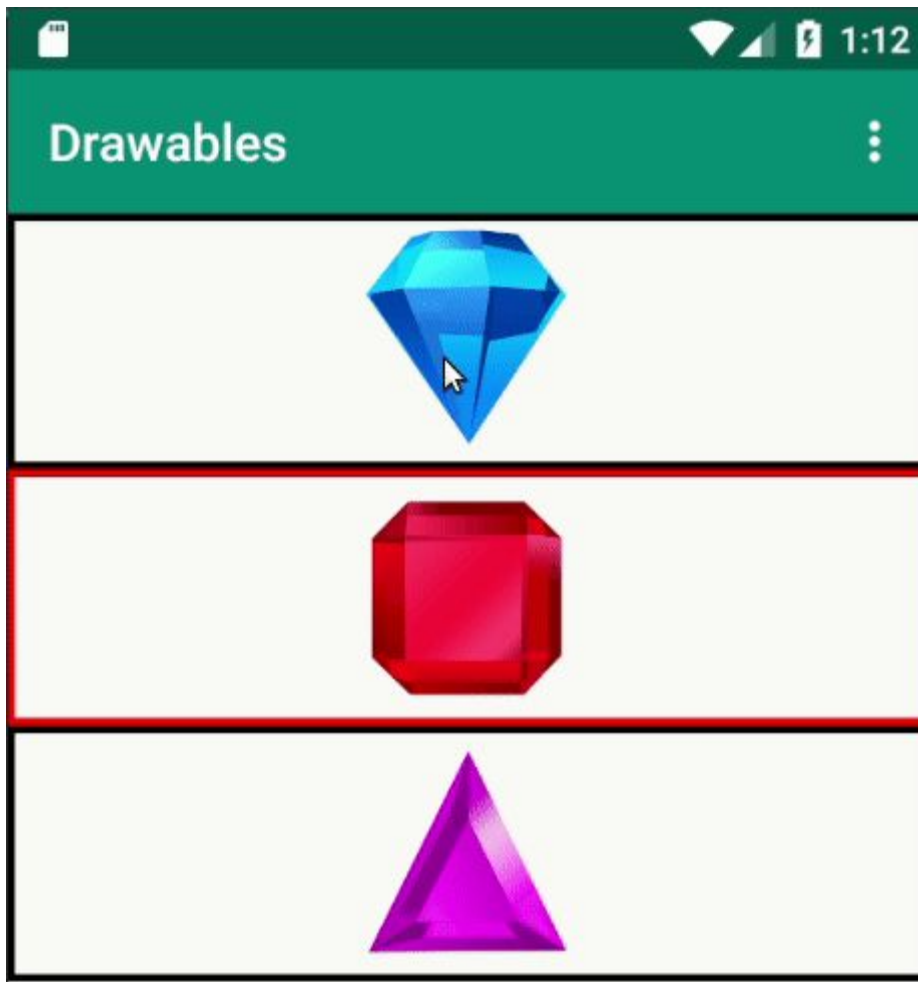
<https://developer.android.com/guide/topics/resources/drawable-resource?hl=es-419#StateList>

8. Intente utilizar el drawable StateList definido en el ejercicio anterior en otra componente visual como un ImageView o un TextView. ¿Funciona el cambio de estado al presionar sobre la componente?.

Establezca el atributo “clickable” de la componente en true y vuelva a intentarlo.

9. Defina un layout con tres imágenes dispuestas de manera vertical. Establezca un drawable para el fondo de cada imagen de manera que se muestre un borde en color negro. Utilice el estado *focused* de un drawable StateList de manera que cuando el usuario haga foco presionando sobre una de las imágenes, la misma muestre el borde en rojo mientras que las demás queden en negro.

Nota: Para que funcione el estado *focused* de una componente visual, hay que asegurarse que los atributos *clickable*, *focusable* y *focusableInTouch* estén en true.



10. Modifique el ejercicio anterior para utilizar el estado *selected* en lugar de *focused*.

Nota: Deberá modificar manualmente el estado de la componente utilizando el método *setSelected*.