

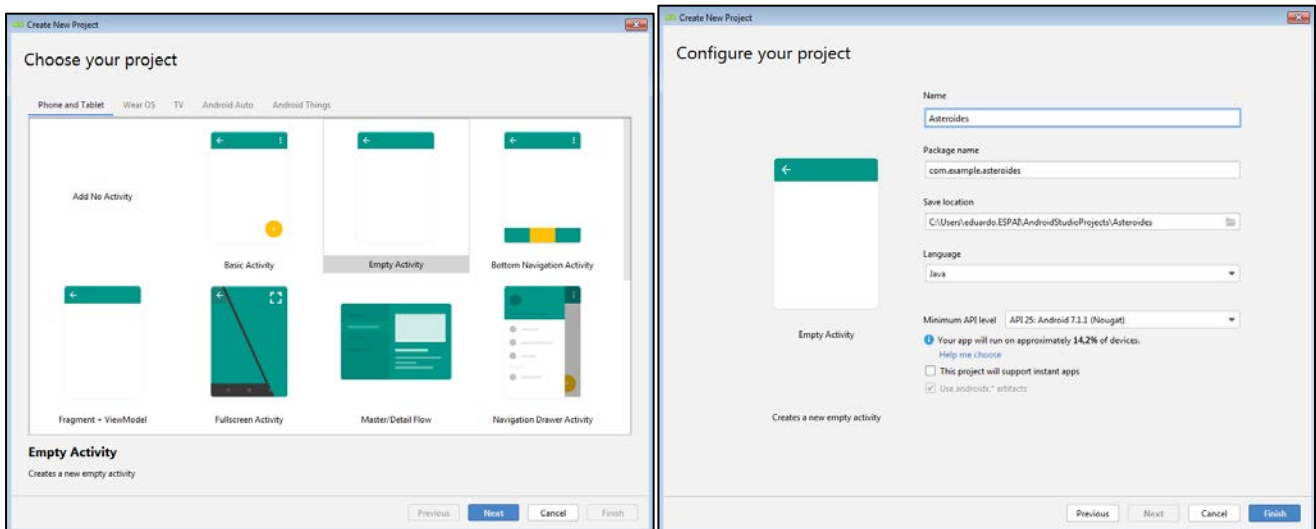
## PRACTICA 10: ASTEROIDES I

### Parte I: Asteroides

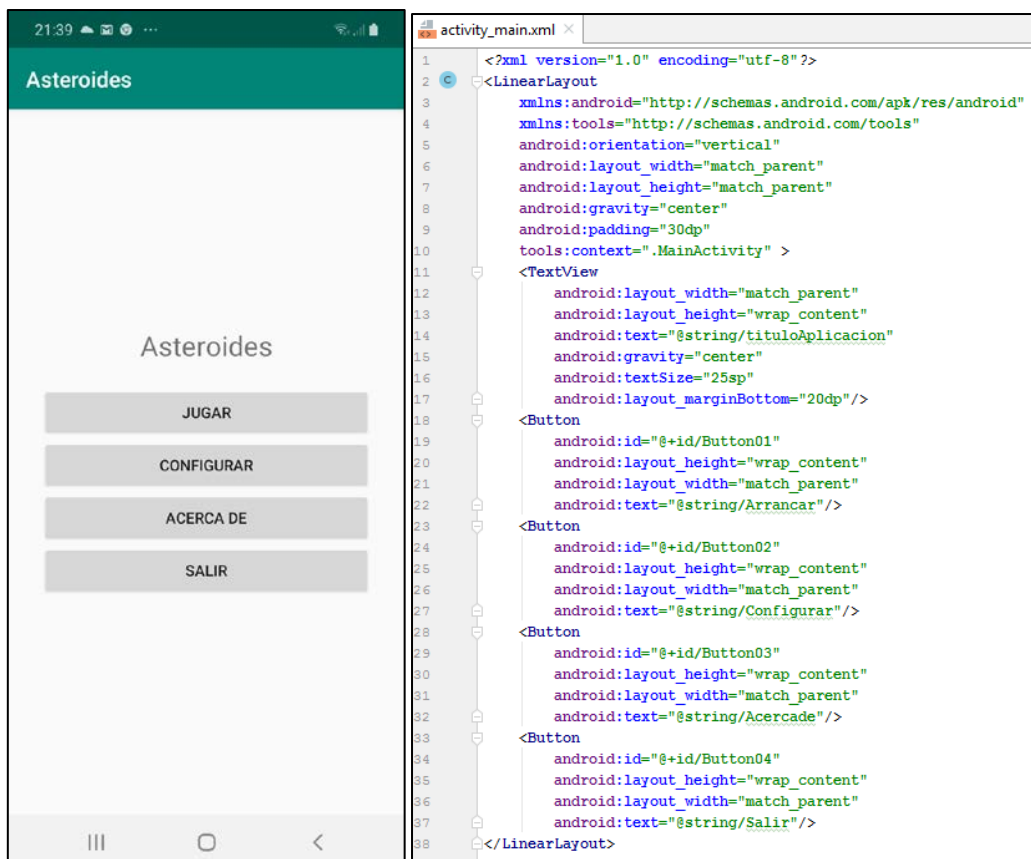
**Paso 1.** Crea un nuevo proyecto, de nombre Asteroides, con los siguientes datos:

- *Application Name:* Asteroides
- *Add an activity:* Empty View Activity

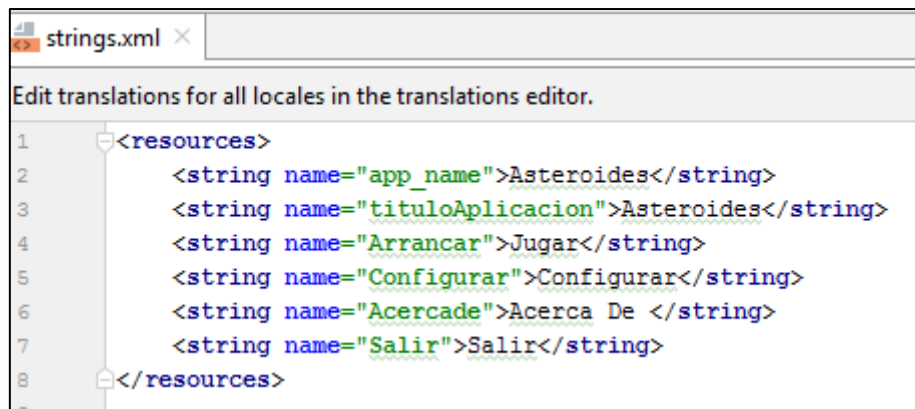
**NOTA:** Deja el resto de los parámetros con su valor por defecto.



**Paso 2.** Abre el fichero `res/Layout/main.xml` y trata de crear una vista similar a la que ves a continuación. Ha de estar formada por un `LinearLayout` que contiene un `TextView` y cuatro `Button`. Trata de utilizar recursos para introducir los cinco textos que aparecen.



**Paso 3.** El fichero `res/values/strings.xml` ha de tener el siguiente contenido:




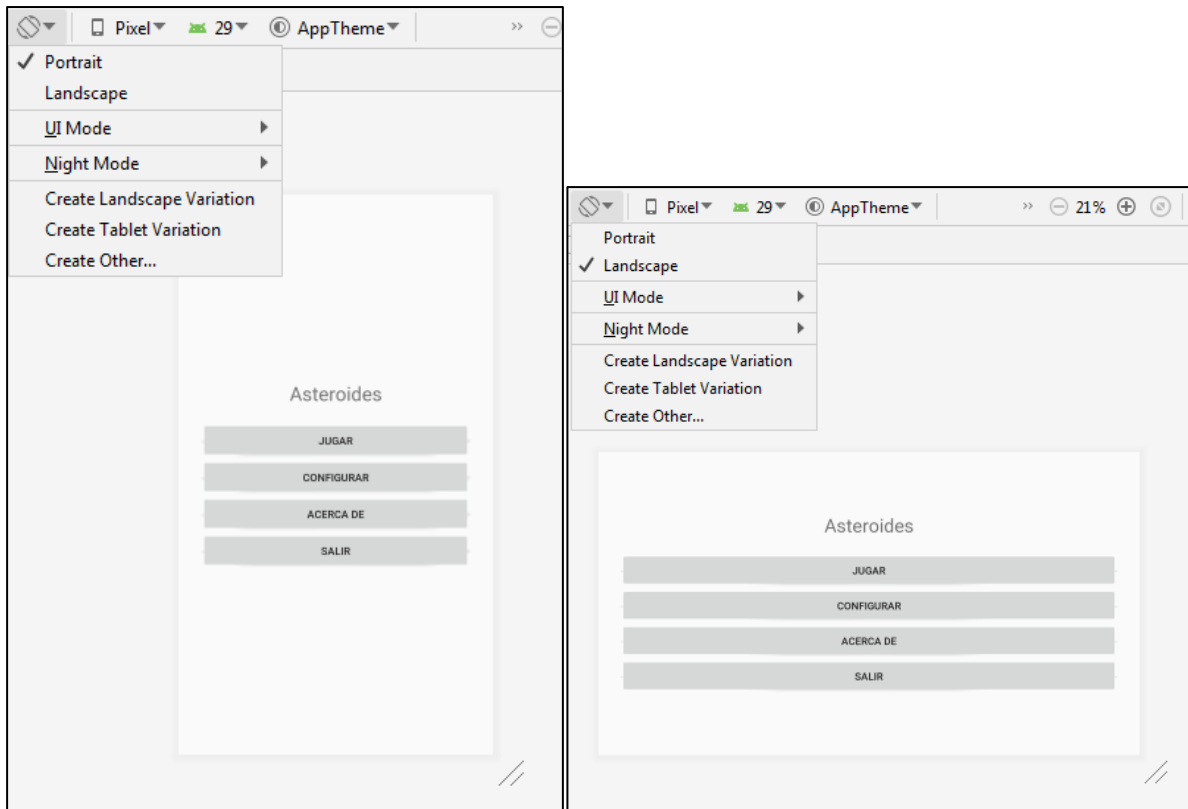
```
1 <resources>
2     <string name="app_name">Asteroides</string>
3     <string name="tituloAplicacion">Asteroides</string>
4     <string name="Arrancar">Jugar</string>
5     <string name="Configurar">Configurar</string>
6     <string name="Acercade">Acerca De </string>
7     <string name="Salir">Salir</string>
8 </resources>
```

## Parte II: Asteroides

**Paso 1.** Ejecuta la aplicación creada en el punto anterior en el emulador.

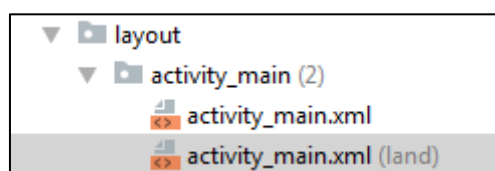
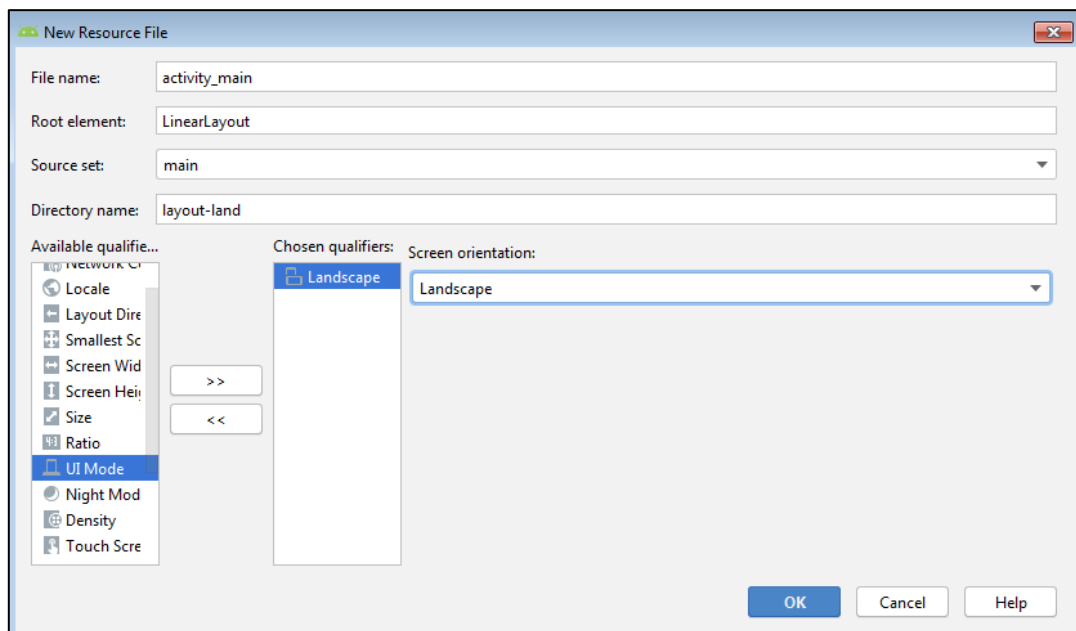


**Paso 2.** Los teléfonos móviles basados en Android permiten cambiar la configuración en apaisado y en vertical. Para conseguir este efecto con el emulador pulsa el botón . Si, usas un dispositivo con pantalla pequeña, puedes observar que el resultado de la vista que acabas de diseñar en vertical, no queda todo lo bien que deseáramos.



Para resolver este problema Android te permite diseñar una vista diferente para la configuración horizontal y otra para vertical.

**Paso 3.** Pulsa con el botón derecho sobre la carpeta `res/layout` y selecciona `New > Layout resource file`. Aparecerá una ventana donde has de rellenar en `File name:content_main`, en `Available qualifiers:` selecciona **Orientation** y pulsa en el botón `>>`. En el desplegable `Screen orientation:` selecciona `Landscape`. Pulsa en `OK`. Observa como ahora hay dos recursos para el fichero `content_main.xml`. El primero es el recurso por defecto mientras que el segundo es el que se usará cuando el dispositivo esté en orientación `Landscape`.



**Paso 4.** Crea en el nuevo layout (land) una vista similar a la que ves a continuación: formada por un *TableLayout* con dos *Button* por columna. C.



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:orientation="vertical"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:gravity="center"
9      android:padding="30dp"
10     tools:context=".MainActivity" >
11     <TextView
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:text="@string/tituloAplicacion"
15         android:gravity="center"
16         android:textSize="25sp"
17         android:layout_marginBottom="20dp"/>
18     <TableLayout
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:gravity="center"
22         android:stretchColumns="*">
23         <TableRow>
24             <Button android:id="@+id/button01"
25                 android:layout_height="wrap_content"
26                 android:layout_width="match_parent"
27                 android:text="@string/Arrancar"/>
28             <Button android:id="@+id/button02"
29                 android:layout_height="wrap_content"
30                 android:layout_width="match_parent"
31                 android:text="@string/Configurar"/>
32         </TableRow>
33         <TableRow>
34             <Button android:id="@+id/button03"
35                 android:layout_height="wrap_content"
36                 android:layout_width="match_parent"
37                 android:text="@string/Acercade"/>
38             <Button android:id="@+id/button04"
39                 android:layout_height="wrap_content"
40                 android:layout_width="match_parent"
41                 android:text="@string/Salir"/>
42         </TableRow>
43     </TableLayout>
44 </LinearLayout>

```

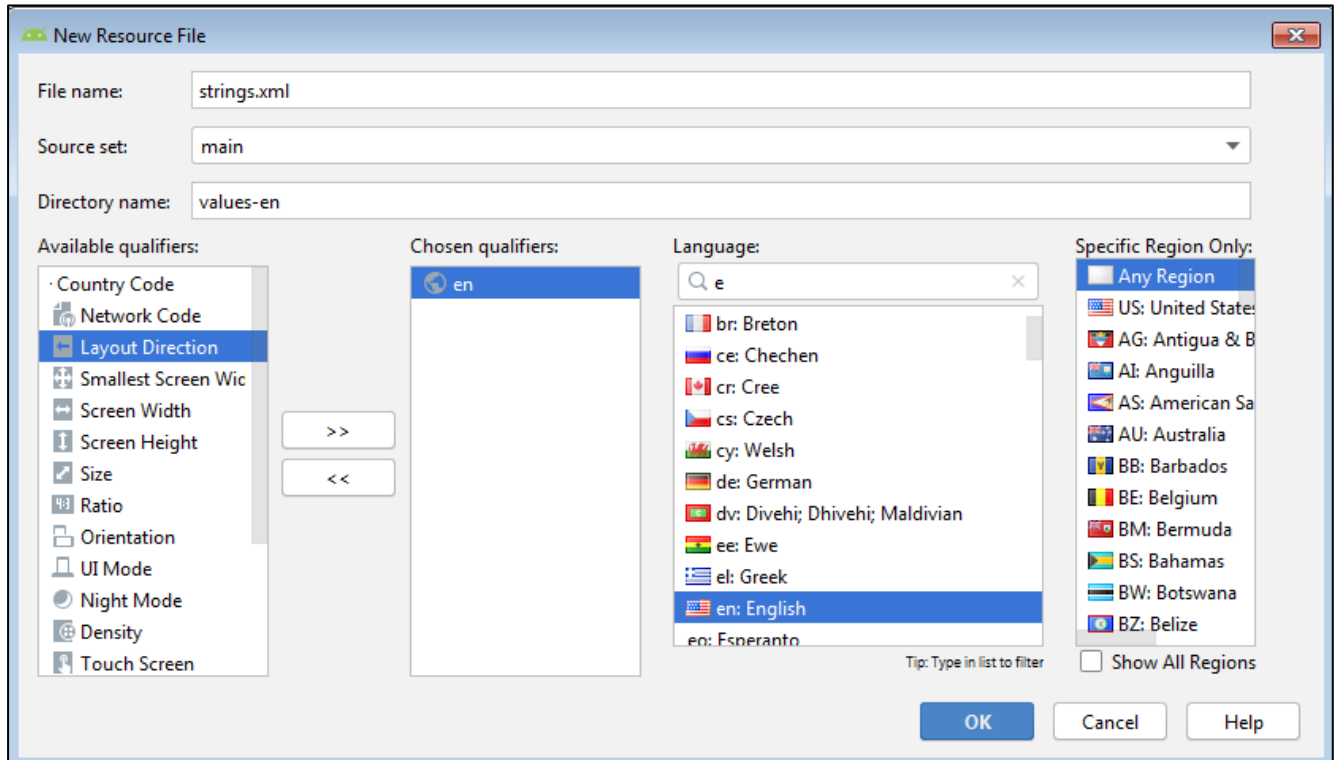
**NOTA:** Para conseguir en un *TableLayout*, que las columnas se ajusten a todo el ancho de la tabla poner *stretchColumns="\*"*. *stretchColumns="0"* significa que asigne el ancho sobrante a la primera columna. *stretchColumns="1"* significa que asigne el ancho sobrante a la segunda columna. *stretchColumns="\*\*"* significa que asigne el ancho sobrante entre todas las columnas.

**Paso 5.** Ejecuta de nuevo la aplicación y observa como la vista se ve correctamente en las dos orientaciones



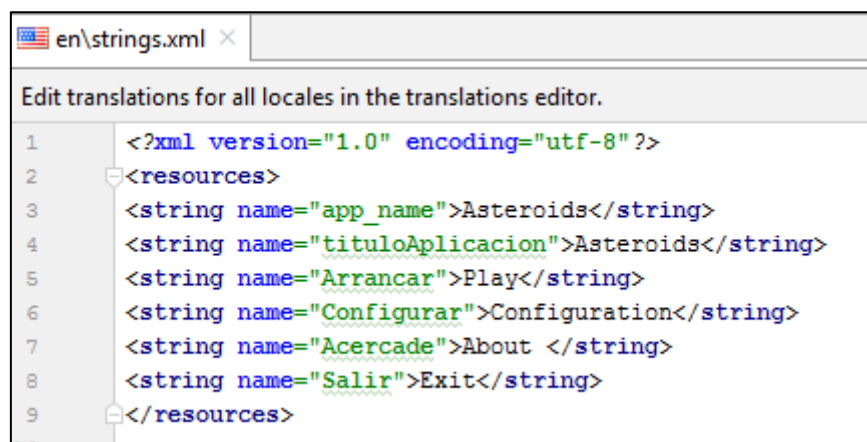
### Parte III: Asteroides

**Paso 1.** Crea un nuevo recurso alternativo para strings.xml (en): Pulsa con el botón derecho en res/values, selecciona New > Values resource file e introduce strings.xml como nombre de fichero. Mueve el cualificador **Locale** desde el marco de la izquierda a la derecha, pulsando el botón >>. En Language selecciona en: English, en Specific Region Only deja el valor Any Region Only y pulsa OK.

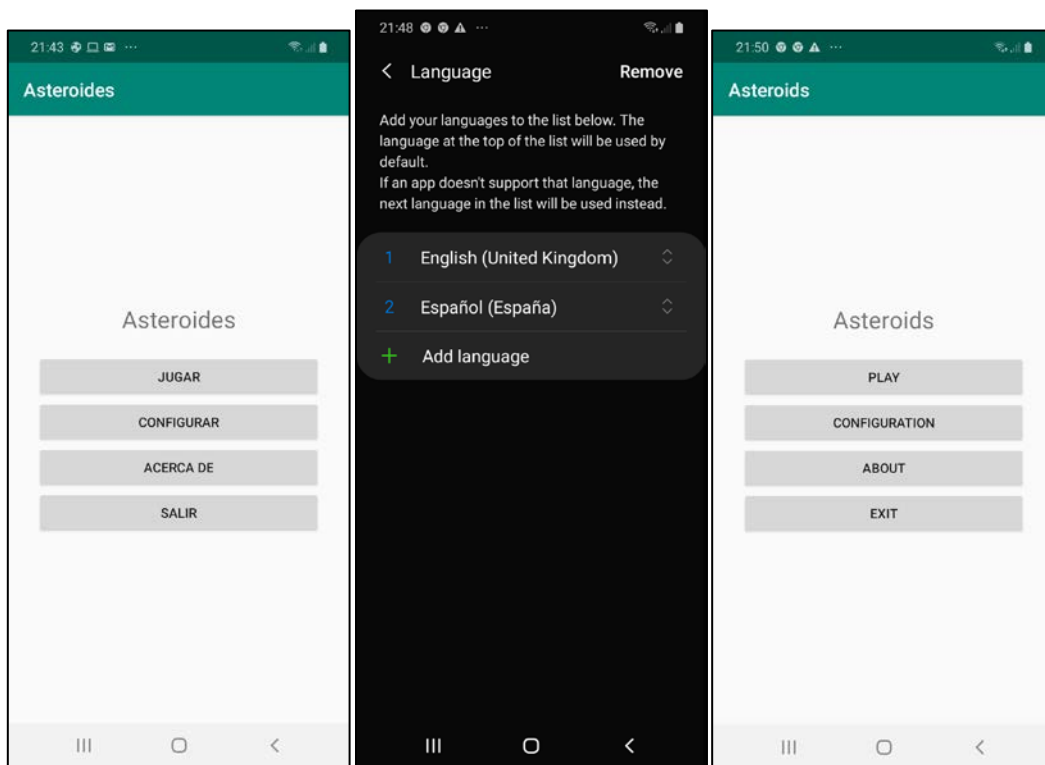


NOTA: Observa cómo además del idioma también permite seleccionar la región. De esta forma podremos diferenciar *entre inglés americano, británico, australiano, ...*

**Paso 2.** Copia el contenido del recurso por defecto, strings.xml, al recurso para inglés, strings.xml (en). Traduce los textos al inglés. No has de traducir los nombres de los identificadores de recursos (accion\_mostrar, app\_name, ...) estos han de ser igual en todos los idiomas.



**Paso 3.** Ejecuta la aplicación.



**Paso 4.** Vamos a cambiar la configuración de idioma de un dispositivo Android. Para ello, accede a Ajustes del dispositivo (Settings) y selecciona la opción Mi dispositivo > Idioma e introducción. Dentro de esta opción selecciona como idioma Español o Inglés.

**NOTA:** Observa que en otros idiomas permite seleccionar tanto el idioma como la región. Por desgracia, para el español solo permite dos regiones: España y Estados Unidos.

**Paso 5.** Ejecuta de nuevo la aplicación y observa cómo ha traducido el texto.

#### **Parte IV: Asteroides**

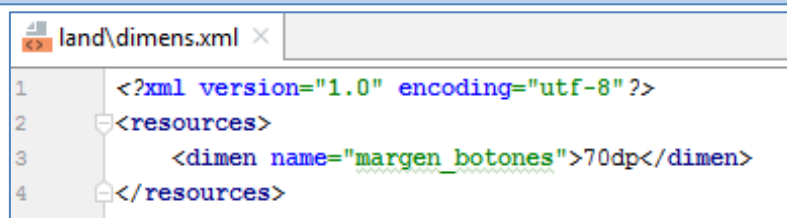
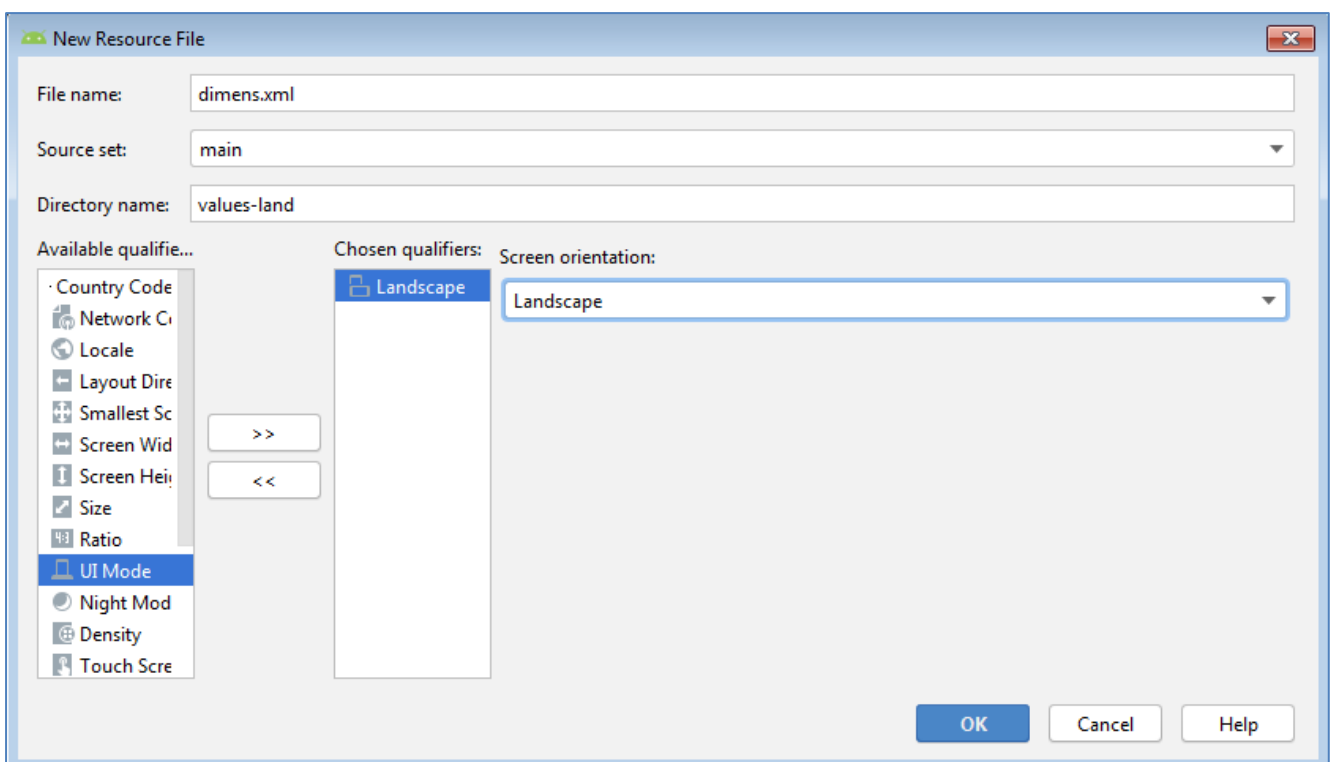
Si ejecutas la aplicación Asteroides en una tableta observarás que los botones son demasiado alargados:



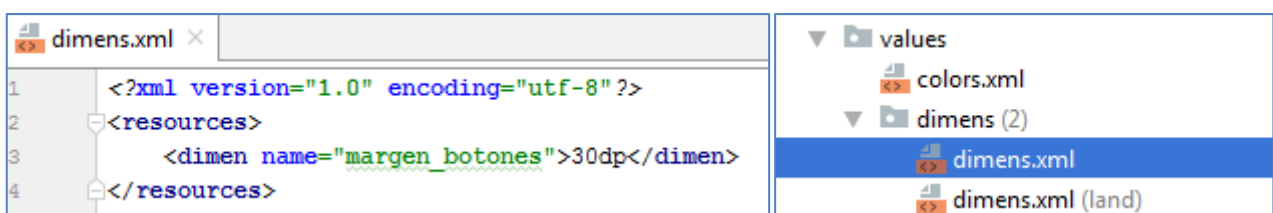
Queremos que en este caso la apariencia sea similar a:



**Paso 1.** Crea un recurso alternativo a `res/values/dimens.xml`, que sea utilizado en pantallas de tamaño (size) `xlarge` (7-10,5 pulgadas) en orientación (orientation) `land` (apaisado). En este fichero define el siguiente valor:



**Paso 2.** Añade el mismo valor al recurso por defecto, pero esta vez con 30dp.



**Paso 3.** Modifica los ficheros `res/layout/activity_main.xml` y `activity_main.xml (land)`, reemplazando `android:padding="30dp"` por `android:padding="@dimen/margen_botones"`.

```

activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:orientation="vertical"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:gravity="center"
9      android:padding="@dimen/margen_botones"
10     tools:context=".MainActivity" >

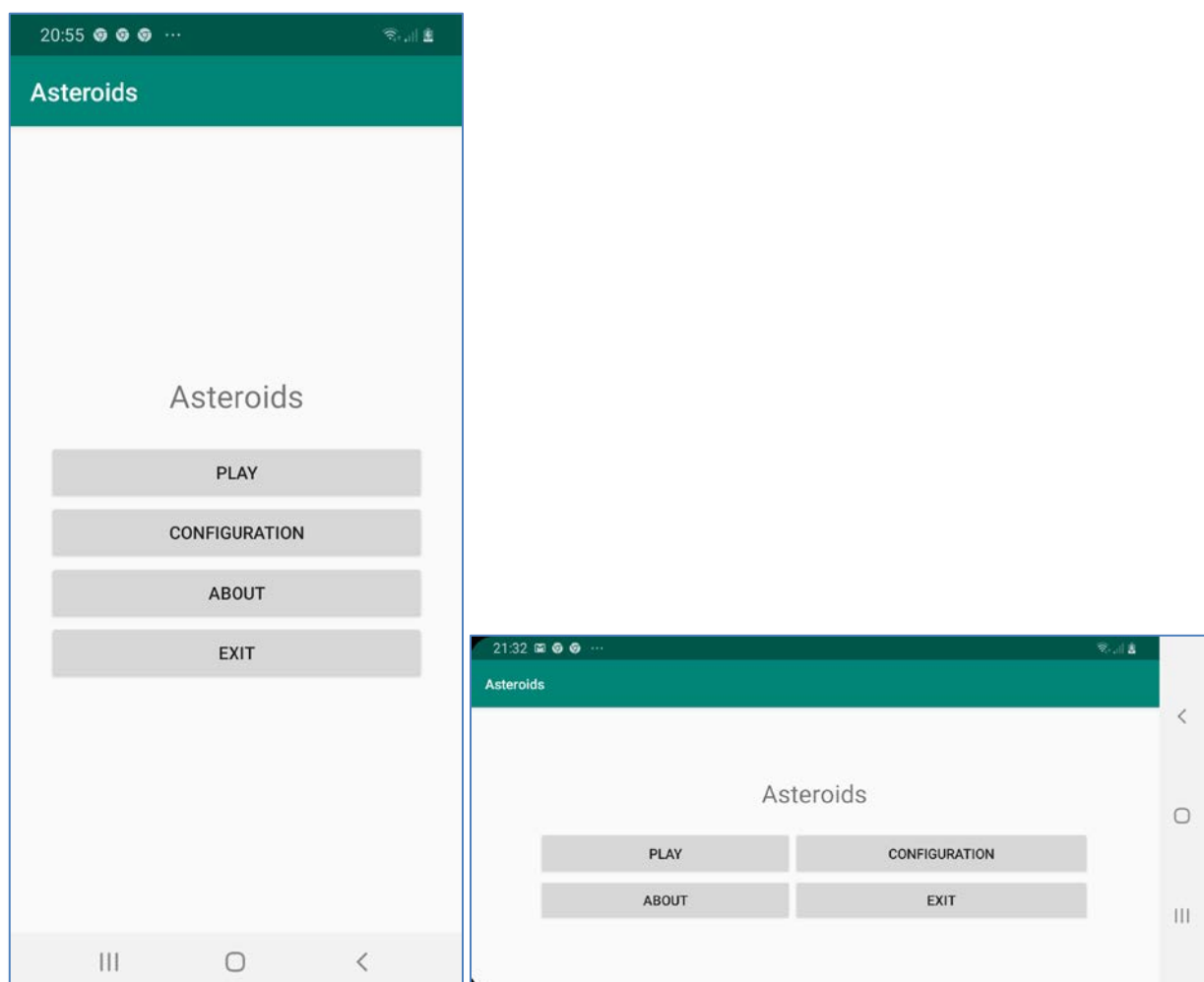
```

```

land\activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:orientation="vertical"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:gravity="center"
9      android:padding="@dimen/margen_botones"
10     tools:context=".MainActivity" >

```

**Paso 4.** Verifica que la aplicación se visualiza correctamente en todos los tipos de pantalla, tanto en horizontal como en vertical.





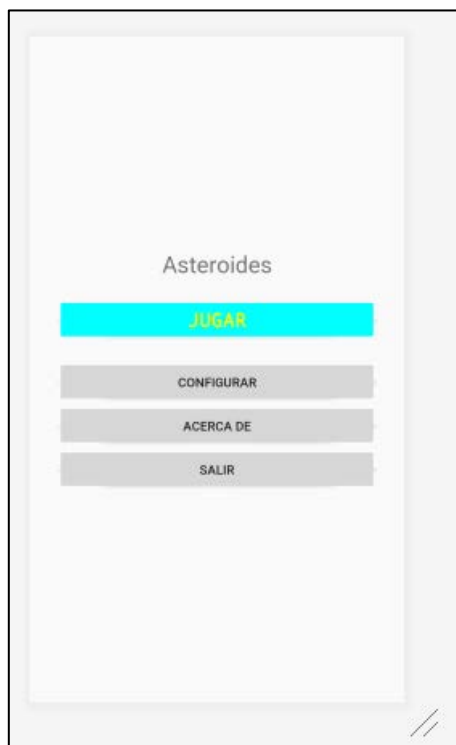
## Parte V: Asteroides. Creando un estilo en Asteroides

**Paso 1.** Abre el proyecto Asteroides. Crea un nuevo estilo con nombre EstiloBoton. Para ver las propiedades que puedes modificar te recomendamos que consultes la "Referencia de la clase View". Para un botón puedes definir los atributos de View, TextView y Button. Otra alternativa consiste en seleccionar un botón en el editor visual de vistas y en la ventana Properties buscar las propiedades disponibles:

Attributes		
Button01		
editorExtras		0
elegantTextHeight		0
elevation		0
ellipsize		0
ems		0
enabled		0
fadeScrollbars		0
fadingEdge		0
fadingEdgeLength		0
fallbackLineSpacing		0
filterTouchesWhenObscured		0
firstBaselineToTopHeight		0
fitsSystemWindows		0
focusable	true	0
focusableInTouchMode		0
fontFamily	sans-serif-medium	0
fontFeatureSetting		0
fontVariationSetting1		0
forceHasOverlappingRendering		0
foreground		0
foregroundGravity		0
foregroundTint		0

```
<style name="EstiloBoton" parent="@android:style/TextAppearance.Medium">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:textColor">#FFFF00</item>
    <item name="android:typeface">monospace</item>
    <item name="android:textSize">20dp</item>
    <item name="android:textStyle">bold</item>
    <item name="android:backgroundTint">#00FFFF</item>
    <item name="android:layout_marginBottom">20dp</item>
    <item name="android:gravity">center</item>
</style>
```

**Paso 2.** Aplícalo al primer botón del layout.



```
<Button
    android:id="@+id/Button01"
    style="@style/EstiloBoton"
    android:text="@string/Arrancar"/>
```

**Paso 3.** Crea un nuevo estilo con nombre EstiloBoton.Alternativo. Este ha de modificar alguno de los atributos anteriores y añadir otros, como padding.

```
<style name="EstiloBoton.Alternativo" parent="@android:style/TextAppearance.Medium">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:textColor">#FF00FF</item>
    <item name="android:typeface">monospace</item>
    <item name="android:textSize">20dp</item>
    <item name="android:textStyle">bold</item>
    <item name="android:backgroundTint">#00FF00</item>
    <item name="android:layout_marginBottom">20dp</item>
    <item name="android:gravity">center</item>
    <item name="android:padding">20dp</item>
</style>
```

**Paso 4.** Aplícalo al segundo botón del Layout.

```
<Button
    android:id="@+id/Button01"
    style="@style/EstiloBoton"
    android:text="@string/Arrancar"/>

<Button
    android:id="@+id/Button02"
    style="@style/EstiloBoton.Alternativo"
    android:text="@string/Configurar"/>
```

**Paso 5.** Visualiza el resultado.



## **Parte VI: Asteroides. Aplicando un tema del sistema**

**Paso 1.** Abre el proyecto Asteroides. Aplica a la actividad principal el tema `@style/Theme.AppCompat.Dialog` tal y como se acaba de mostrar. Este tema hace que nuestra aplicación aparezca como una caja de dialogo

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Asteroides"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.Dialog">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

**Paso 2.** Visualiza el resultado. Este tema es utilizado en cuadros de dialogo. No parece muy adecuado para nuestra actividad.



**Paso 3.** Deshaz el cambio realizado en este ejercicio

## Parte VII: Asteroides. Creando un tema en Asteroides

**Paso 1.** Abre el proyecto Asteroides. Abre el fichero `res/values/styles.xml` (recurso por defecto). Observa cómo se define el estilo `AppTheme` que será usado como estilo por defecto en la aplicación. Hereda de `Theme.AppCompat.Light.DarkActionBar` y solo define los colores principales usados en la aplicación. Añade las líneas subrayadas para personalizar un par de aspectos:

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

    <item name="android:typeface">serif</item>
    <item name="android:textColor">#0000FF</item>

    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
</resources>
```

**Paso 2.** Ejecuta la aplicación para visualizar el resultado.



**Paso 3.** Modifica otros atributos y comprueba el resultado.