Cómo implementar interfaces de usuario adaptables

En función del diseño actual de tu aplicación, la interfaz puede variar. Por ejemplo, si tu aplicación está en modo de panel dual, haz clic en un elemento del panel izquierdo para que aparezca en el panel de la derecha. Asimismo, si está en modo de panel único, el contenido debería aparecer por sí mismo (en otra actividad).

EN ESTA SECCIÓN PUEDES APRENDER:

1. Cómo determinar el diseño actual
2. Cómo reaccionar en función del diseño actual
3. Cómo volver a utilizar fragmentos en otras actividades
4. Cómo gestionar los cambios en la configuración de la pantalla

TAMBIÉN PUEDES CONSULTAR:

- Cómo admitir tablets y dispositivos móviles

¡PRUÉBALO!

Descargar la aplicación de ejemplo

NewsReader.zip

Cómo determinar el diseño actual

Dado que la implementación de cada diseño variará en cierta medida, probablemente una de las primeras cosas que tendrás que hacer será determinar el diseño que el usuario puede ver en ese momento. Por ejemplo, puedes determinar si el usuario utiliza el modo de "panel único" o de "panel dual". Para ello, puedes consultar si una vista determinada existe y es visible:

```
public class NewsReaderActivity extends Fragment
    boolean mIsDualPane;

    @Override
    public void onCreate(Bundle savedInstanceSta
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_layout);

        View articleView = findViewById(R.id.art
        mIsDualPane = articleView != null &&
                      articleView.getVisibilit
    }
}
```

Ten en cuenta que este código consulta la disponibilidad del panel del "artículo", lo que es mucho más flexible que incluir una consulta para un diseño determinado.

Otro ejemplo de cómo puedes adaptar la existencia de diferentes componentes es comprobar su disponibilidad antes de realizar una operación relacionada con los mismos. Por ejemplo, en la aplicación de ejemplo News Reader, hay un botón que abre un menú, pero ese botón solo aparece en las versiones anteriores a Android 3.0 (porque `ActionBar (/reference/android/app/ActionBar.html)` en el nivel 11 del API y en niveles superiores). Por tanto, para añadir el detector de eventos para este botón, puedes hacer lo siguiente:

```
Button catButton = (Button) findViewById(R.id.categorybutton);
OnClickListener listener = /* create your listener here */;
if (catButton != null) {
    catButton.setOnClickListener(listener);
}
```

Cómo reaccionar en función del diseño actual

El resultado de algunas acciones puede variar en función del diseño actual. Por ejemplo, en el ejemplo de News Reader, al hacer clic en un encabezado de la lista se abrirá el artículo del panel situado a la derecha si la interfaz de usuario está en modo de panel dual, pero se iniciará una actividad independiente si esta está en modo de panel único:

```
@Override
public void onHeadlineSelected(int index) {
    mArtIndex = index;
    if (mIsDualPane) {
        /* display article on the right pane */
        mArticleFragment.displayArticle(mCurrentCat.getArticle(index));
    } else {
        /* start a separate activity */
        Intent intent = new Intent(this, ArticleActivity.class);
        intent.putExtra("catIndex", mCatIndex);
        intent.putExtra("artIndex", index);
        startActivity(intent);
    }
}
```

De igual modo, si la aplicación está en modo de panel dual, debe configurar la barra de acción con pestañas para la navegación, mientras que si la aplicación está en modo de panel único, debe configurar la navegación con un widget giratorio. Por tanto, el código debe comprobar también cuál es el caso adecuado:

```
final String CATEGORIES[] = { "Top Stories", "Politics", "Economy", "Technology" };

public void onCreate(Bundle savedInstanceState) {
    ....
    if (mIsDualPane) {
        /* use tabs for navigation */
        actionBar.setNavigationMode(android.app.ActionBar.NAVIGATION_MODE_TABS);
        int i;
        for (i = 0; i < CATEGORIES.length; i++) {
            actionBar.addTab(actionBar.newTab().setText(
                CATEGORIES[i]).setTabListener(handler));
        }
        actionBar.setSelectedNavigationItem(selTab);
    }
    else {
        /* use list navigation (spinner) */
        actionBar.setNavigationMode(android.app.ActionBar.NAVIGATION_MODE_LIST);
        SpinnerAdapter adap = new ArrayAdapter(this,
                R.layout.headline_item, CATEGORIES);
        actionBar.setListNavigationCallbacks(adap, handler);
    }
}
```

Cómo volver a utilizar fragmentos en otras actividades

Un patrón recurrente a la hora de diseñar para distintas pantallas es utilizar una parte de la interfaz que se implementa como un panel en algunas configuraciones de pantalla y como actividad independiente en otras. Por ejemplo, en el ejemplo de News Reader, el texto del artículo de noticias se presenta en el panel de la derecha en las pantallas grandes, pero es una actividad independiente en las pantallas más pequeñas.

En otros casos similares, puedes evitar la duplicación de código reutilizando la misma `Fragment` (/reference/android/app/Fragment.html) en distintas actividades. Por ejemplo, `ArticleFragment` se utiliza en el diseño de panel dual:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
```

```
    <fragment android:id="@+id/headlines"
              android:layout_height="fill_parent"
              android:name="com.example.android.newsreader.HeadlinesFragment"
              android:layout_width="400dp"
              android:layout_marginRight="10dp"/>
    <fragment android:id="@+id/article"
              android:layout_height="fill_parent"
              android:name="com.example.android.newsreader.ArticleFragment"
              android:layout_width="fill_parent" />
</LinearLayout>
```

A continuación, se vuelve a utilizar (sin diseño) en el diseño de actividad para pantallas más pequeñas (`ArticleActivity`):

```
ArticleFragment frag = new ArticleFragment();
getSupportFragmentManager().beginTransaction().add(android.R.id.content, frag).commit(
```

Evidentemente, esto tiene el mismo efecto que declarar el fragmento en un diseño XML. Sin embargo, en este caso, un diseño XML sería un trabajo innecesario porque el fragmento del artículo es el único componente de esta actividad.

Un punto muy importante que debes tener en cuenta al diseñar tus fragmentos es no crear un acoplamiento fuerte para una actividad determinada. Para ello, normalmente puedes definir una interfaz que resuma todas las formas en las que tiene que interactuar el fragmento con su actividad principal y, a continuación, la actividad principal implementa esa interfaz:

Por ejemplo, ese es precisamente el objetivo del `HeadlinesFragment` de la aplicación News Reader:

```
public class HeadlinesFragment extends ListFragment {
    ...
    OnHeadlineSelectedListener mHeadlineSelectedListener = null;

    /* Must be implemented by host activity */
    public interface OnHeadlineSelectedListener {
        public void onHeadlineSelected(int index);
    }
    ...

    public void setOnHeadlineSelectedListener(OnHeadlineSelectedListener listener) {
        mHeadlineSelectedListener = listener;
    }
}
```

A continuación, cuando el usuario selecciona un encabezado, el fragmento notifica el detector especificado por la actividad principal (en lugar de notificar una actividad predefinida específica):

```
public class HeadlinesFragment extends ListFragment {
    ...
    @Override
    public void onItemClick(Adapt     ew<?> parent,
                            View ...., int position, long id) {
        if (null != mHeadlineSelecte Listener) {
            mHeadlineSelectedLis     .onHeadlineSelected(position);
        }
    }
    ...
}
```

Si quieres obtener más información sobre esta técnica, puedes consultar la guía sobre Cómo admitir tablets y dispositivos m[illegible]ets.html).

Cómo ges[illegible]a configuración de la pantalla

Si utilizas acti[illegible]nentar partes individuales de tu interfaz, debes tener en cuenta que es [illegible]nte determinados cambios en la configuración (como un cambio de rota[illegible]dad de tu interfaz.

Por ejemplo, er[illegible]3.0 o una versión superior, el ejemplo de News Reader utiliza una actividad i[illegible]lo de noticias en el modo vertical, pero utiliza el diseño de panel dual en e[illegible]

Esto significa [illegible]lo vertical y está consultando un artículo, tienes que detectar que la orientac[illegible]l y reaccionar de forma adecuada finalizando la actividad. A continuación, [illegible] para que el contenido pueda mostrarse en el diseño de panel dual:

```
public cla[illegible] FragmentActivity {
    int mC[illegible]

    @Overr[illegible]
    protec[illegible] vedInstanceState) {
        su[illegible] tate);
        mC[illegible] tras().getInt("catIndex", 0);
        mA[illegible] tras().getInt("artIndex", 0);

        // If should be in two-pa[illegible] ode, finish to return to main activity
        if (getResources().getBo[illegible] (R.bool.has_two_panes)) {
            finish();
            return;
        }
        ...
}
```