

Desarrollo de aplicaciones móviles con Android



Fragmentos

Un Fragment representa un comportamiento o una parte de la interfaz de usuario en una Activity. Puedes combinar múltiples fragmentos en una sola actividad para crear una IU multipanel y volver a usar un fragmento en múltiples actividades.

Puedes pensar en un fragmento como una sección modular de una actividad que tiene su ciclo de vida propio, recibe sus propios eventos de entrada y que puedes agregar o quitar mientras la actividad se esté ejecutando (algo así como una "subactividad" que puedes volver a usar en diferentes actividades).

Fragmentos

Un fragmento siempre debe estar integrado a una actividad y el ciclo de vida del fragmento se ve directamente afectado por el ciclo de vida de la actividad anfitriona.

Por ejemplo, cuando la actividad está pausada, también lo están todos sus fragmentos, y cuando la actividad se destruye, lo mismo ocurre con todos los fragmentos. Sin embargo, mientras una actividad se está ejecutando (está en el estado del ciclo de vida reanudada), puedes manipular cada fragmento de forma independiente; por ejemplo, para agregarlos o quitarlos. Cuando realizas una transacción de fragmentos como esta, también puedes agregarlos a una pila de actividades administrada por la actividad; cada entrada de la pila de actividades en la actividad es un registro de la transacción de fragmentos realizada. La pila de actividades le permite al usuario invertir una transacción de fragmentos (navegar hacia atrás) al presionar el botón Atrás.

Fragmentos

Cuando agregas un fragmento como parte del diseño de tu actividad, este se ubica en ViewGroup, dentro de la jerarquía de vistas de la actividad y el fragmento define su propio diseño de vista. Puedes insertar un fragmento en el diseño de tu actividad declarando el fragmento en el archivo de diseño de la actividad como elemento `<fragment>` o agregándolo a un ViewGroup existente desde el código de tu aplicación. Sin embargo, no es necesario que un fragmento forme parte del diseño de la actividad; también puedes usar un fragmento con su IU propia, como un trabajador invisible para la actividad.

Filosofía de diseño

Android introduce los fragmentos en Android 3.0 (nivel de API 11), principalmente para admitir diseños de IU más dinámicos y flexibles en pantallas grandes, como las de las tablets. Como la pantalla de una tablet es mucho más grande que la de un teléfono, hay más espacio para combinar e intercambiar componentes de la IU.

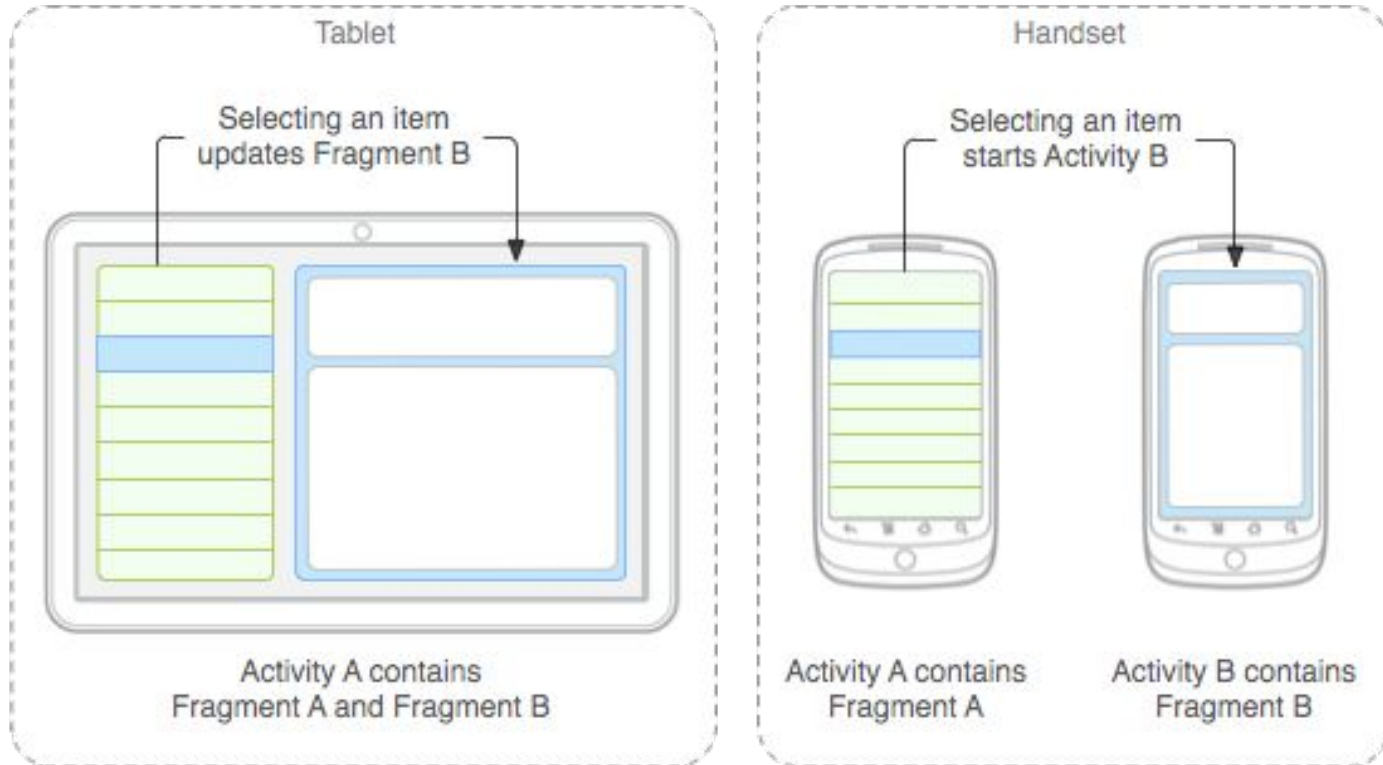
Los fragmentos admiten esos diseños sin la necesidad de que administres cambios complejos en la jerarquía de vistas. Al dividir el diseño de una actividad en fragmentos, puedes modificar el aspecto de la actividad durante el tiempo de ejecución y conservar esos cambios en una pila de actividades administrada por la actividad.

Filosofía de diseño

Cada fragmento debe ser un componente modular y reutilizable de la actividad. Como cada fragmento define su propio diseño y su propio comportamiento con sus propios callbacks del ciclo de vida, entonces se puede incluir un fragmento en múltiples actividades.

Esto es muy importante porque un fragmento modular te permite cambiar tus combinaciones de fragmentos para diferentes tamaños de pantalla. Cuando diseñas tu aplicación para que admita tablets y teléfonos, puedes reutilizar tus fragmentos en diferentes configuraciones de diseño para optimizar la experiencia del usuario en función del espacio de pantalla disponible.

Fragmento



Crear un fragmento

Para crear un fragmento, debes crear una subclase `Fragment` (o una subclase existente de ella). La clase `Fragment` tiene un código que se asemeja bastante a una `Activity`. Contiene métodos callback similares a los de una actividad, como

`onCreate()`

`onStart()`

`onPause()`

`onStop()`.

`onCreate()`. El sistema lo llama cuando crea el fragmento. En tu implementación, debes inicializar componentes esenciales del fragmento que quieres conservar cuando el fragmento se pause o se detenga y luego se reanude.

`onCreateView()`. El sistema lo llama cuando el fragmento debe diseñar su interfaz de usuario por primera vez. Para diseñar una IU para tu fragmento, debes devolver una `View` desde este método que será la raíz del diseño de tu fragmento. Puedes devolver nulo si el fragmento no proporciona una IU.

Crear un fragmento

`onPause()`

El sistema llama a este método como el primer indicador de que el usuario está abandonando el fragmento (aunque no siempre significa que el fragmento se esté destruyendo). Generalmente este es el momento en el que debes confirmar los cambios que deban conservarse más allá de la sesión de usuario actual (porque es posible que el usuario no vuelva).

La mayoría de las aplicaciones deben implementar al menos estos tres métodos para cada fragmento, pero hay muchos otros métodos callback que también debes usar para manipular varias fases del ciclo de vida del fragmento. Todos los métodos callback del ciclo de vida se discuten en más detalle más adelante en la sección Manipulación del ciclo de vida del fragmento.

