Module VII
Android Fragment

- Fragment
- Dialog
- ToolBar(Action Bar)
- Tabs

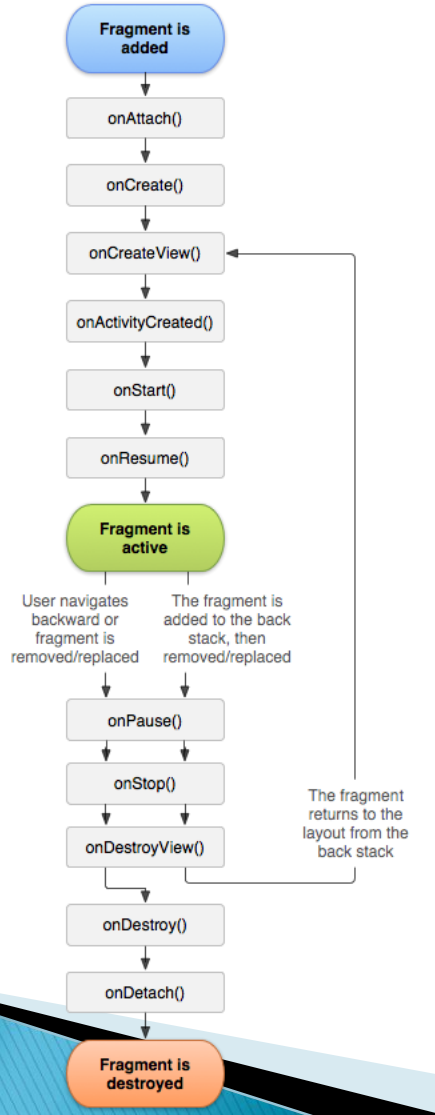
Fragment (1/4)

- 因為Android裝置尺寸眾多，加上解析度也都不盡相同，因此有可能造成小螢幕的UI設計放到大螢幕上會不合適。所以需要一個具有高度彈性的設計機制，讓應用程式能依照執行環境而自動調整操作介面

- Android 3.0 (API level 11)開始加入了Fragment類別，讓我們可以將畫面分割成多個區域，能各自隱藏或顯示：
  - 程式執行畫面可以由多個Fragment組成
  - 每個Fragment都有各自獨立的執行狀態，並接收各自處理的事件
  - 程式執行過程中，能動態加入或移除Fragment
  - Fragment需依附在Activity，無法獨立存在

- 為了讓Fragment技術支援Android3.0以前版本，使用android.support.v4.app.Fragment開發即可讓Android1.6以上的系統支援Fragment

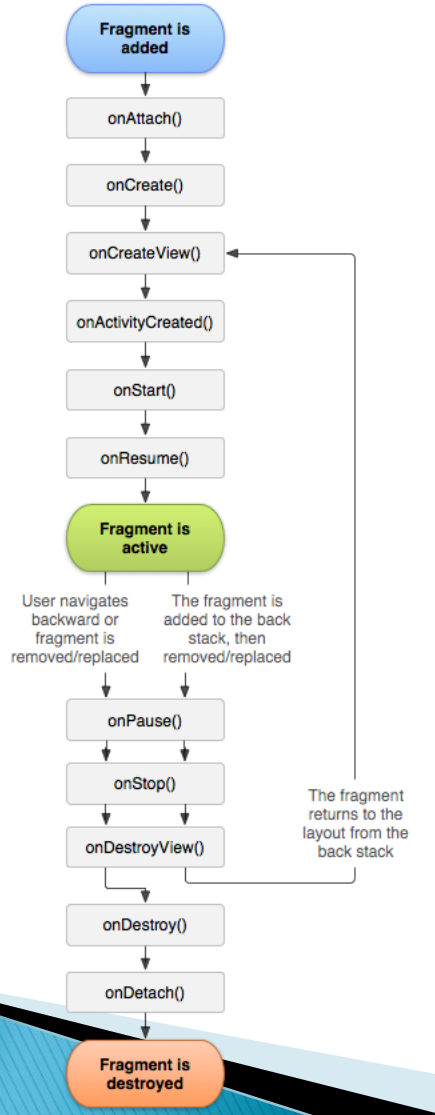
Fragment (2/4)



從Fragment附加到Activity上，到呈現畫面給使用者時，會經歷以下方法：

- onAttach()：Fragment第一次附加在Activity時會呼叫此方法
- onCreate()：在此初始化Fragment
- onCreateView()：在此初始化Fragment的UI元件
- onActivityCreated()：呼叫此方法代表所屬的Activity已建立完畢(即Activity.onCreate()執行完成)
- onStart()：Fragment畫面準備要呈現
- onResume()：Fragment畫面可與使用者開始互動

Fragment (3/4)



從Fragment畫面準備離開，到脫離Activity時，會經歷以下方法：

- onPause()：當Activity進入暫停狀態(即Activity.onPause())或Fragment準備要脫離Activity時，即呼叫此方法
- onStop()：當Activity進入停止狀態(即Activity.onStop())或Fragment因脫離Activity而停止時，即呼叫此方法
- onDestroyView()：Fragment畫面確定脫離Activity時呼叫此方法，通常在此方法內釋放與清除跟Fragment畫面有關的資源
- onDestroy()：呼叫此方法代表Fragment不再被使用
- onDetach()：Fragment確定完全脫離Activity

Fragment (4/4)

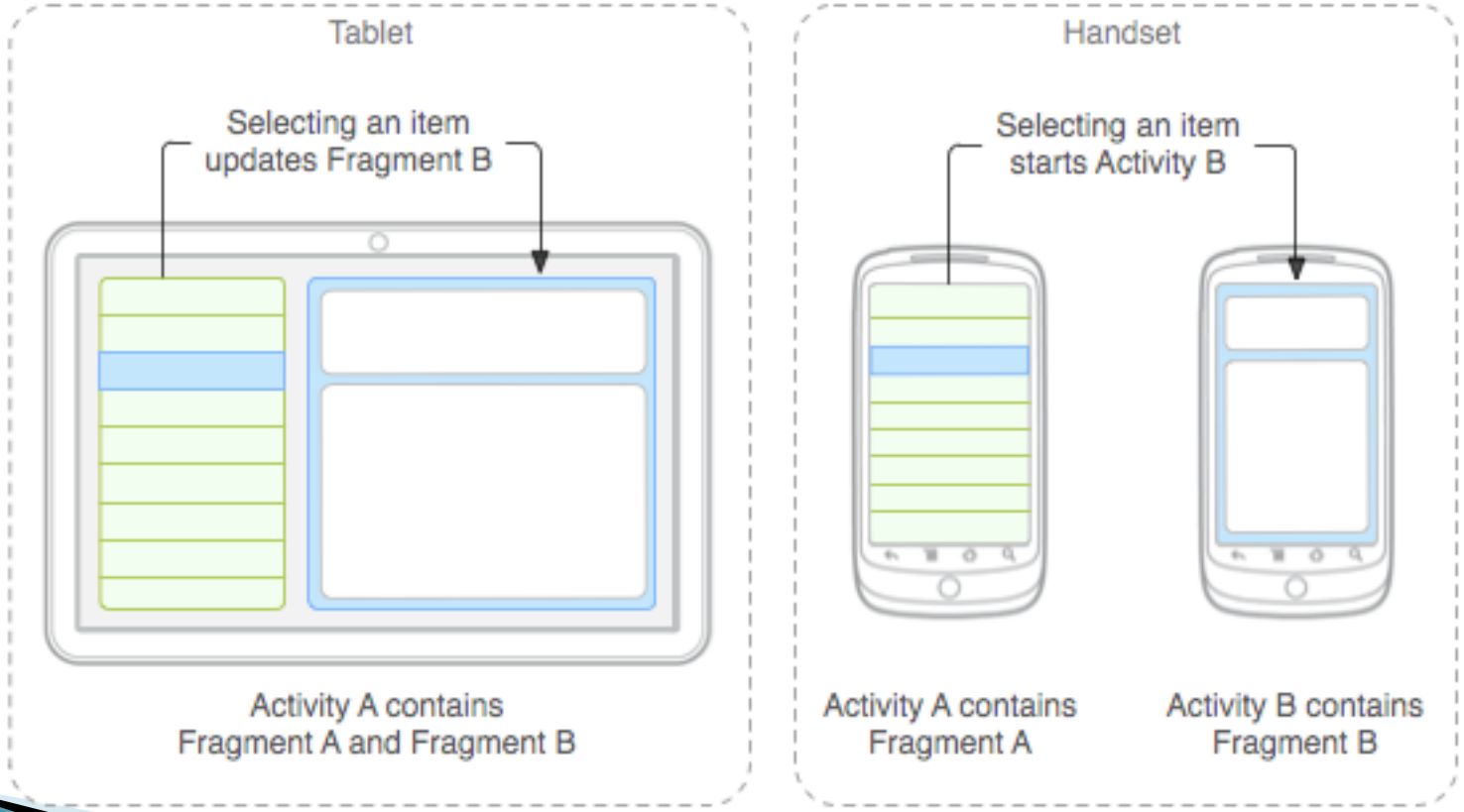
- 靜態配置Fragment：
  - 在layout檔案使用<fragment>標籤，並在android:name屬性指定對應的完整類別路徑即可

- 動態配置Fragment：
  - 呼叫getFragmentManager()方法取得FragmentManager物件
  - 呼叫FragmentManager.beginTransaction()方法取得FragmentTransaction物件
  - 透過FragmentTransaction物件進行add, replace等方法
  - 設定完畢後呼叫commit()方法進行更新

範例：FragmentEx

Flexible Fragment UI

- 為對應裝置螢幕大小顯示適合操作的畫面，我們可以將Activity切割，每個畫面都由不同的Fragment處理



範例：FlexibleFragmentEx

Alert Dialog

- AlertDialog是Dialog的子類別，最常看見與使用的就是對話視窗

- 建立AlertDialog時，必須設定3個部份：
  - 對話視窗的標題文字
  - 對話視窗的訊息文字
  - 對話視窗的按鈕與文字與各自對應的事件處理

範例：AlertDialogEx

Date/Time Picker Dialog

- 產生Date Picker Dialog對話視窗步驟如下
  - 改寫DialogFragment.onCreateDialog()方法以提供內容
  - 實作OnDateSetListener.onDateSet()，日期挑選器的「設定」鈕被按下就會呼叫此方法
  - 呼叫DialogFragment.show()以呈現對應的Dialog

- Time Picker Dialog產生步驟與方式與Date Picker Dialog非常類似，請參考範例

範例：DateTimePickerDialogEx

Custom Dialog

- 前面所提到對話視窗，如Date/Time Picker Dialog與之後會介紹的ProgressDialog等都是已經具備特定功能且使用方式已固定，若是要自訂自己專用的對話視窗，就必須使用到Dialog類別

- 建立步驟如下：
  - 建立自訂對話視窗的layout檔
  - 建立一個Dialog類別的物件
  - 設定標題、載入layout檔與設定Cancelable屬性
  - 建立對話視窗裡面各元件Listener與相關事件處理
  - 呼叫show()方法顯示

範例：CustomDialogEx

Action Bar (1/3)

- Action Bar為Android 3.0(API level 11)後才推出的新技術，讓使用者能更方便地體驗APP。若是要讓Android 2.X也能支援Action Bar，就必須使用support library提供的API
- 提供了Button, Spinner與Tab等功能讓使用者能輕鬆地操作
- 適用於各種寬窄畫面，Action Bar會依畫面自動調整顯示方式，若是開發的應用程式想要平板手機通吃，Action Bar可以善加利用

Action Bar (2/3)

- Action Bar說明(續)：
  (1) 擺設應用程式圖示
  (2) Action按鈕
  (3) 延伸列(Action overflow)：畫面裝不下的就以Menu鍵方式呈現

- Action Bar重要的相關屬性showAsAction：
  - always：該選項總是放置於Action Bar上面
  - ifRoom：若當下Action Bar還有空間就放置，否則就隱藏
  - never：該選項不放置在Action Bar上面
  - withText：該選項只呈現文字部份並恢復為傳統Options Menu
  - collapseActionView：讓選項與Options Menu相同呈現方式

ToolBar (3/3)

- 從Android 5.0 (API level 21)以後，原Action Bar類別不少方法已被官方Deprecated，這代表Google希望能使用ToolBar取代Action Bar
- 使用時需注意套用的theme是否有ActionBar，如果有，需要將XML屬性windowActionBar設為false，windowNoTitle設為true
- 若是要支援5.0以下的版本，在layout檔加入的ToolBar元件必須是android.support.v7.widget.ToolBar
- 在程式碼中先透過findViewById取得ToolBar物件後，再呼叫setSupportActionBar(ToolBar toolBar)即可建立

範例：ToolBarEx

ViewPager

- 與ListView非常相似，可以達到整頁切換的功能
- 建立FragmentStatePagerAdapter子類別並改寫getCount()與getItem()方法
- 呼叫setAdapter，將FragmentStatePagerAdapter設定傳入即可
- 記憶體只會保留畫面當前的Fragment，這對要切換大量頁面的程式來說，是個很好的設計方式

範例：ViewPagerEx

Tabs

- 用標籤的分式呈現分頁，引導使用者點選
- 搭配Fragment一起使用，可參考範例
- 進階版可再搭配ViewPager做到用手勢即可滑動切換Tabs的功能！

範例：TabsEx