RecyclerView + CardView 效果

原始碼：https://github.com/stone0705/RecyclerViewANDCardView

RecyclerView:

基本介紹： RecyclerView為ListView的進階版本，主要用於表現大量的資料List。

特色：會回收畫面外的項目重新使用在現在的畫面，以增進效能。

用法與功能：

使用RecyclerView主要需要設置：

必須：

(1).LayoutManager：

LayoutManager主要用於控制RecyclerView的Layout控制，可以使用LayoutManager來達成版面的配置。

LayoutManager分為三種：

1.LinearLayoutManager：



(左為垂直版面設置，右為水平版面設置)

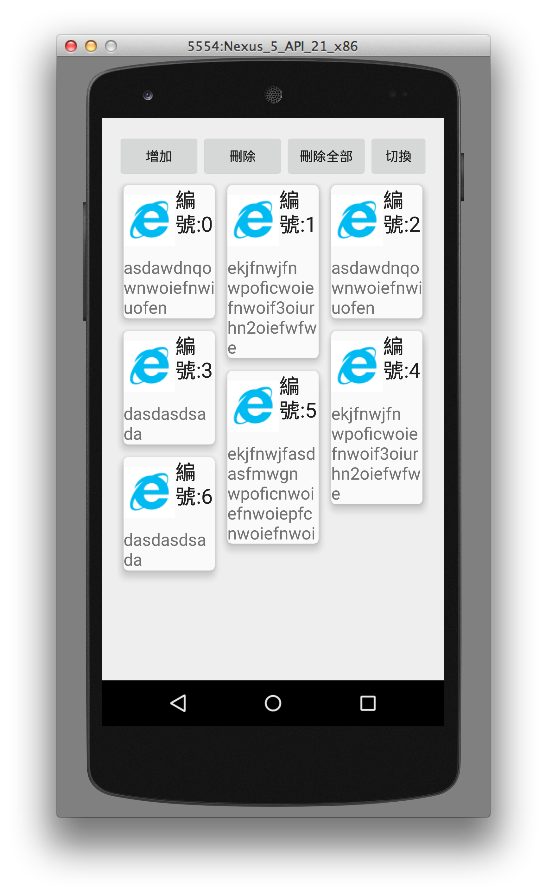
LinearLayoutManger用於實現普通LinearLayout的版面，就像普通的選單效果。

2.GridLayoutManager ：



(左為垂直版面設置，右為水平版面設置)

GridLayoutManager用於實現方格式版面設置，圖為以每行或每列3格設計。

3.StaggeredGridLayoutManager：

(左為垂直版面設置，右為水平版面設置)

StaggeredGridLayoutManager用於實現不規則長度的方格式版面設置，圖為每行或每列3格設計。

(2).RecyclerView.ViewHolder：

由於後面要使用Adapter所以ViewHolder要先實作。

ViewHolder主要工作就是控制RecyclerView裏面所要包含的view。

實作ViewHolder要實作包含super建構子的建構子。

ViewHolder主要負責宣告view裡面的元件如Button,TextView等，在建構子中使用findViewByID來註冊元件。

(3).RecyclerView.Adapter：

Adapter 主要控制資料儲存

RecyclerView.Adapter裡面包含幾個abstract方法，需要自己實作：

1. onCreateViewHolder

public ViewHolder onCreateViewHolder(ViewGroup viewGroup,int i)

在我們實作完ViewHolder後，需要在這個方法中指定ViewHolder所要使用的layout的xml檔案，並把它回傳。

2.onBindViewHolder

public void onBindViewHolder(ViewHolder viewHolder,int position)

這個方法主要是在RecyclerView畫面上出現物品時呼叫的方法，如捲動捲軸或新增物品等，

它所使用的參數viewHolder和position就是新出現的那個物品，所以你在出來的﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽出現其他物品或是 position實作ViewHolder時所宣 告的Button,TextView等都可以在這邊設定Listener跟文字等。

需要注意的部分為這個方法為回收之前使用的物品再使用，如果之前有改變過高度或寬度，可見或不可見等，後來Bind出來的也會跟著影響。

3.getItemCount

public int getItemCount()

此方法為取得Adapter的資料數量，此方法實作方式基於你資料儲存的方式。

可選：

(1).ItemAnimator:

主要用於呈現RecyclerView裡面物品加入移除移動等動畫。

Android提供預設的動畫為：

DefaultItemAnimator

預設動畫基本上呈現淡出和淡入效果。

使用預設動畫時加入物品時adapter要使用notifyItemInserted 移除物品使用notifyItemRemoved才會出現動畫效果。

(2).ItemDecoration:

用於繪製RecyclerView中物品與物品中間的分隔線，預設沒有繪製，需自行實作繪圖部分。

(4).ItemTouchHelper:

這個能夠有效幫助你完成drag跟swipe的實現

CardView:

基本介紹：能夠呈現卡片班的外觀

特色：帶有圓角跟立體陰影。

用法與功能：

主要能做的功能大致上有：

(1).設定圓角的半徑

(2).設定陰影的高低

(3).設定卡片和裡面內容的間隔(內容距離卡片外圍的距離)

Demo程式：

這個程式主要分為三個部分：

共通按鈕為：增加，刪除，刪除全部，切換

增加為添加隨機內容。

刪除為刪除隨機內容。

刪除全部為刪除全部內容。

切換為切換版面配置(big跟expand 主要是針對LinearLayout的版面設計 不建議切換)

(1).linear

較為基本的做法，RecycleView裡面只包含一種類型的Item，按切換可更換LayoutManager配置，按X可切換水平垂直配置。

按下會轉到另一個Activity不建議切換﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽。

(2).big

比linear多設置了按下會展開功能，CardView多設置了卡片與內容之間的距離。

(3).expand

這跟big一樣按下會展開，裡面實作方式與big不同。

(4).drag

比linear多設置了drag和swipe功能

程式實作部分：

(1).linear：

主頁面在LinearActivity。

RecyclerView的Adapter在LinearAdapter。

裡面包含了adapter,viewholder,item

item裡面包含了2個String,1個int，int儲存圖片的ID。

Viewholder包含了2個textview,1個imageview,1個cardview。

建構子註冊view的id。

Adapter包含了item的arraylist做為dataset，onbind時將dataset中的值跟listener放進去。

(2).big ：

主頁面在bigActivity。

RecyclerView的Adapter在bigAdapter。

與linear差別在於adapter的onBindViewHolder上，big的cardview裡面比linear的多了一個textview，big的做法是先把它隱藏點開來之後搭配動畫彈出來，由於是使用動畫來改變大小，在 onBindViewHolder時要先將它設定成原本大小，以避免新的card出來時大小不同的問題。

(3).expand：

主頁面在expandableActivity。

expandableActivity的Adapter在expandableAdapter裡面。

Big的做法是將隱藏的顯示出來。

Expand的做法是做出2個不同的view，點下parent 的會展開child的部分。

具體上來說expand的dataset的樣子是這樣

1.parent

2.parent

3.parent

當按下第二個parent 時會在它底下加上它的child於是dataset就變成這樣

1.parent

2.parent

3.child

4.child

5.child

6.parent

這樣就能很簡單的展現出清單式的配置，重點在於parent的margin只對上、左、右作間隔，c

child只對下、左、右作間隔，於是parent就能很好的跟child結合起來。

在實作上的部分 你必須要宣告2個ViewHolder跟2個Item，因為你實際上是有2個view，在adapter中你要實作的部分有：

1.getItemViewType

public int getItemViewType(int position)

這個方法用來決定你在這個位置所要選擇的view是哪一個，回傳的數字可以用一個常數定義好。

2. onCreateViewHolder

public ViewHolder onCreateViewHolder(ViewGroup viewGroup,int i)

在這裡的int參數就是你getItemViewType所取得的參數，你可以用switch或if用這個int來選擇你要的view的xml

3. onBindViewHolder

public void onBindViewHolder(ViewHolder viewHolder,int position)

在這裏你可以對viewHolder或資料做instanceof來判斷這是哪一個view進一步選擇要做的行為。

4.expand

這是我用來展開parent的方法，在parent底下再加上他自己的child。

(4).drag

主頁面在dragdropAcitivity。

Adapter在dragdropAdapter裡面，viewholder跟item我直接使用linear那邊的，adapter本身只比原本的多出2個方法：

1.onItemMove

void onItemMove(int fromPosition, int toPosition)

這方法我會在拖動的時候呼叫，主要目的在變更位置與修改dataset

2.onItemDismiss

void onItemDismiss(int position)

這方法是在我橫滑item時會呼叫，把item刪除。

拖動與橫滑具體實現則是在SimpleItemTouchHelperCallback中達成。

有幾個方法需要實作：

1.isLongPressDragEnabled

public boolean isLongPressDragEnabled()

這方法直接return true 表示長壓會啟動拖動狀態

2.isItemViewSwipeEnabler

public Boolean isItemLongPressDragEnabled()

這方法也是return true 讓你能啟動橫滑功能

3.getMovementFlags

public int getMovementFlags(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder)

這方法主要是選擇哪些動作會啟動功能

4.onMove

public boolean onMove(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder, RecyclerView.ViewHolder target)

這方法讓就是在移動的時候要呼叫的方法，這時回頭呼叫剛剛在adapter實作的onItenMove

5.onSwiped

public void onSwiped(RecyclerView.ViewHolder viewHolder, int direction)

這方法是在滑動的時候呼叫的，同樣也是回頭呼叫adapter的onItemDismiss