PercentLayout AND DragHelper

程式碼: <https://github.com/stone0705/PercentLayoutANDDragHelper>

PercentLayout為我們實現了百分比類型的佈局，類似CSS的方式。

PercentLayout目前有兩種，PercentRelativeLayout和PercentFrameLayout，基本上來講就是RelativeLayout和FrameLayout加上了百分比配置的功能。

具體而言多出來的屬性有:

layout\_widthPercent

layout\_heightPercent

layout\_marginPercent

layout\_marginLeftPercent

layout\_marginTopPercent

layout\_marginRightPercent

layout\_marginBottomPercent

layout\_marginStartPercent

layout\_marginEndPercent

這幾項都可以用百分比來表達。

widthPercent的百分比會以你手機橫向長度來計算，heightPercent則是用手機直向長度計算。

雖然目前google只有兩種layout有支持百分比，但是可以使用PercentLayoutHelper自己實作需要的layout。﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽r.﷽per﷽agHelper。

DragHelper則是幫我們完成了拖拉動作的進行與監聽，google的DrawerLayout就是使用這個來實現。

具體而言我主要是把想要能拖拉的layout重新extends再實作一個新的layout，而監聽與拖拉等功能都可以在ViewDragHelper.Callback中完成。

可以實現的方法有：

1. public int clampViewPositionHorizontal ([View](https://developer.android.com/reference/android/view/View.html) child, int left, int dx)

用來控制水平的位移

2. public int clampViewPositionVertical ([View](https://developer.android.com/reference/android/view/View.html) child, int top, int dy)

用來控制垂直的位移

3. public int getViewHorizontalDragRange ([View](https://developer.android.com/reference/android/view/View.html) child)

回傳水平能移動的範圍

4. public int getViewVerticalDragRange ([View](https://developer.android.com/reference/android/view/View.html) child)

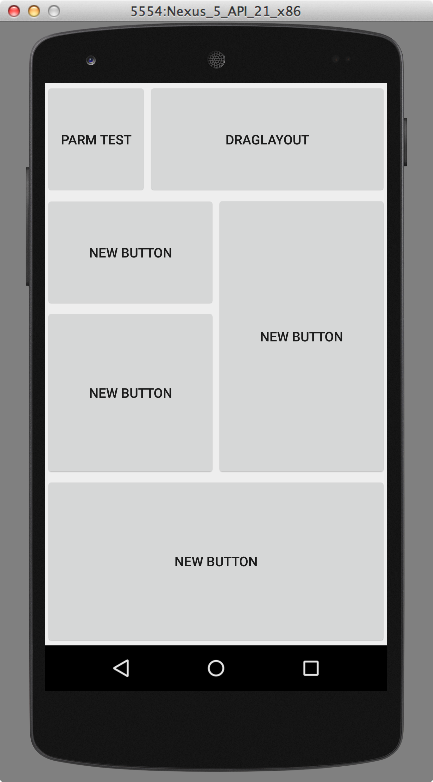
回傳垂直能移動的範圍

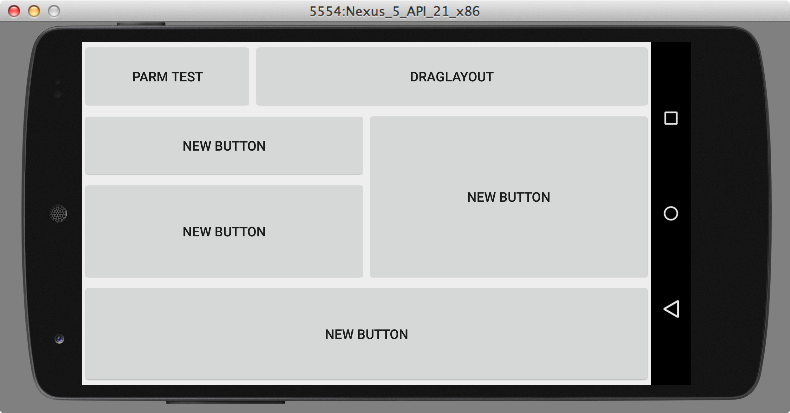
5. public void onEdgeDragStarted (int edgeFlags, int pointerId)

可以使用從邊界移動的功能，DrawerLayout抽屜的拉法

其他可以實現的方法有在抓取時事件跟釋放事件跟位置移動事件等等。

圖片介紹





這個面板配置是使用PercentRelativeLayout完成的畫面，大小都是直接使用百分比完成。

DEMO程式

1.PercentLayout部分：

畫面中6個按鈕，基本上只有上方2個按鈕有作用，左上方的按鈕主要是表現PercentLayout的參數在JAVA中的設定，右上方的按鈕是可以讓你跳到DragLayout的部分。

它的Java部分是在MainActivity，xml在activity\_main。

在xml的部分，我們可以很輕易地使用百分比方式來配置版面，比較要注意的是記得要讓各個物件百分比加起來小於100以免出現奇怪的畫面。

在java的部分想要控制百分比配置的部分的時候，我們可以使用PercentRelativeLayout.LayoutParams，以這種類型的方式更改參數params.getPercentLayoutInfo().widthPercent，借由改變widthPercent之後在set回View上面，就可以完成百分比的更改。

2.DragLayout的部分：

這邊我們先實作出一個DragLayout

Java檔為DragLayout。

DragLayout我們先extends LinearLayout，以LinearLayout為基礎做出Drag的功能，

public static [ViewDragHelper](https://developer.android.com/reference/android/support/v4/widget/ViewDragHelper.html) create ([ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) forParent, float sensitivity, [ViewDragHelper.Callback](https://developer.android.com/reference/android/support/v4/widget/ViewDragHelper.Callback.html) cb)我們使用這個方法來做出一個ViewDragHelper4.﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽挑想要的部分時做百分比加起來小魚，具體功能部分我們在他的參數那個CallBack實作，如同最剛開始所說的我們可以挑想要的部分實作，在這個DEMO程式實作的部分有：

1. tryCaptureView(View child, int pointerId)

這個要return true你才能抓東西

2. clampViewPositionHorizontal(View child, int left, int dx)

這個我們計算出左邊界跟右邊界讓你拉的位置不會超過螢幕外

3. clampViewPositionVertical(View child, int top, int dy)

這個我們計算出上邊界跟下邊界讓你拉的位置不會超過螢幕外

4. getViewHorizontalDragRange(View child)

回傳可拉動的橫向距離範圍

5. getViewVerticalDragRange(View child)

回傳可拉動的垂直距離範圍

6. onViewReleased (View releasedChild, float xvel, float yvel)

放開拉動時觸發的事件，這邊我們設計當釋放的時候會出現一個toast顯示釋放，而左上角的按鈕會回到100,100這個位置。

7. onEdgeDragStarted(int edgeFlags, int pointerId)

從邊界抓取會產生的事件，這個我後面程式碼有加上mDragger.setEdgeTrackingEnabled(ViewDragHelper.EDGE\_LEFT);

所以它會從左邊開始抓取，基本上我是設定他抓取時會移動中間的方塊。

8.onViewCaptured (View capturedChild, int activePointerId)

這個控制在抓取開始時產生的事件，這邊我也簡單的產生一個toast表現。

9. onViewPositionChanged (View changedView, int left, int top, int dx, int dy)

這個控制位置發生移動時的事件，因為它產生的速度很快，所以我直接print在編譯器上面，可以觀察到他的位置。

在完成callout之後，還必須要複寫Layout的onInterceptTouchEvent(MotionEvent event)

跟onTouchEvent(MotionEvent event)才能讓畫面能夠抓取，然後最後複寫computeScroll()，這個才能夠讓物件能夠自己歸位，在我們那個讓左上方按鈕回到100,100那個方法能夠成功的部分。

最後完成了DragLayout後可以在XML中使用。