android平台下使用點九PNG技術

「點九」是andriod平台的應用軟體開發里的一種特殊的圖片形式，文件擴展名為：.9.png

智慧型手機中有自動橫屏的功能,同一幅介面會在隨著手機(或平板電腦)中的方向傳感器的參數不同而改變顯示的方向,在介面改變方向後,介面上的圖形會因為長寬的變化而產生拉伸,造成圖形的失真變形。

我們都知道android平台有多種不同的解析度，很多控制項的切圖文件在被放大拉伸後，邊角會模糊失真。

OK，在android平台下使用點九PNG技術，可以將圖片橫向和縱向同時進行拉伸，以實現在多解析度下的完美顯示效果。

【普通拉伸和點九拉伸效果對比】



對比很明顯，使用點九後，仍能保留圖像的漸變質感，和圓角的精細度。

從中我們也可以理解為什麼叫「點九PNG」，其實相當於把一張png圖分成了9個部分(九宮格)，分別為4個角，4條邊，以及一個中間區域，4個角是不做拉升的，所以還能一直保持圓角的清晰狀態，而2條水 平邊和垂直邊分別只做水平和垂直拉伸，所以不會出現邊會被拉粗的情況，只有中間用黑線指定的區域做拉伸。結果是圖片不會走樣

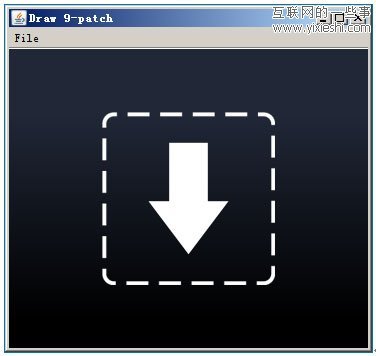
二.「點九」的製作方法

方法1：使用 「draw9patch」工具繪製，流程如下

1. 安裝工具

首先你需要給自己的電腦安裝上java於系統的默認目錄下。沒有安裝java的同學可以在百度搜索：jdk-6u20-windows-i586，安裝包大小80M左右。 一些事

然後使用andriod模擬器—android-sdk-windows，打開SDK/tools目錄下的「draw9patch.bat」文件，出現載入窗口：



2.導入並編輯

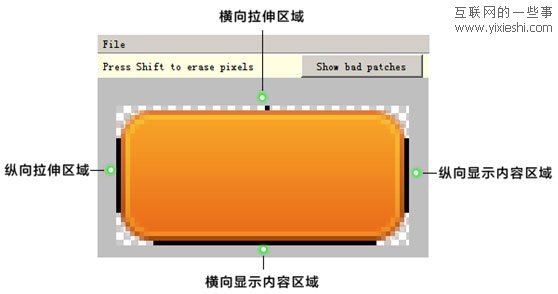
將png圖片拖拽到該窗口中

如下圖，自動進入編輯介面。圖中介紹了每個區域的內容及功能注釋。



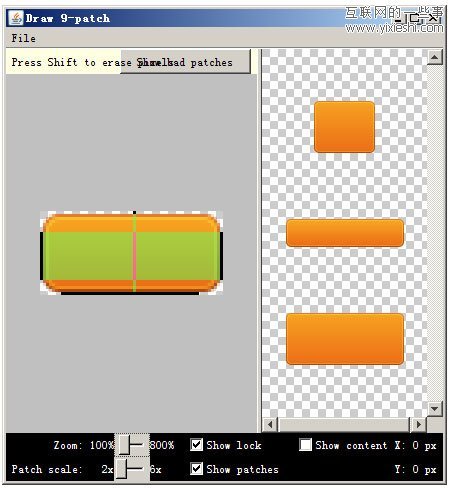
預覽右側的視圖發現，圖片的邊緣處於普通拉伸狀態。

現在我們在圖片邊緣點擊左鍵，繪製出黑線，即圖片需要被拉伸的部分。如下圖，對4條黑線做了注釋。

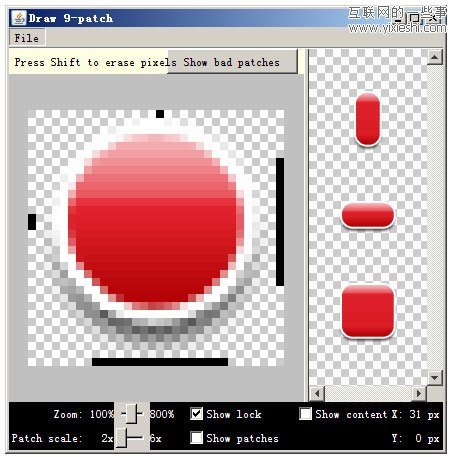
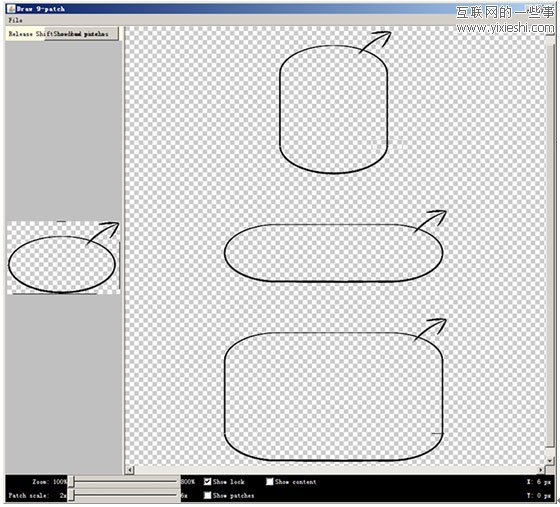


如果失誤多繪的部分，可按住shift鍵的同時點擊滑鼠左鍵擦除)。

如圖所見，三種拉伸結果均已完美顯示，已實現我們想要的拉伸效果，假設這是一個有顯示文字的窗體，那麼文字顯示的區域，程序也會控制在黑線對應範圍。 網際網路的一些事



以下這些圖片，包括異性(非規則圖形)圖片，也可以通過點九PNG實現橫縱向的自然拉伸。 網際網路的一些事



【draw9patch.bat其他功能說明】

② Show lock：顯示不可繪區域

② Show patches：預覽這個繪圖區中的可延伸宮格(粉紅色代表一個可延伸區域)

③ Show patches：預覽視圖中的高亮區域(紫色區域) 一些事

④ Show bad patches：在宮格區域四周增加一個紅色邊界，這可能會在圖像被延伸時產生人工痕跡。如果你消除所有的壞宮格，延伸視圖的視覺一致性將得到維護。

3.保存和輸出 網際網路的一些事

點擊左上file- save，保存文件，自動生成一張後綴名為「\*.9.png」格式的圖片，圖片上下左右各增加了1px的黑線。

方法2：直接使用 PS等平面工具繪製，流程圖如下 yixieshi

如流程圖所示，相對與方法1，只需2個步驟就可得到.9.png圖片，具體步驟為：

1. 確定切圖後直接改變圖片的畫布大小，

2. 手動將上下左右各增加1px

3. 使用鉛筆工具，手動繪製拉伸區域，色值必須為黑色(#000000)。 一些事

4. 存儲為web所用格式，選擇png-24，儲存時手動將後綴名改為.9.png

不過這種方法的缺點是不能實時預覽，判斷並測試拉伸區域的準確性。

使用此方法需要注意以下2點:

1. 手繪的黑線拉伸區必須是#000000，透明度100%，並且圖像四邊不能出現半透明像素;

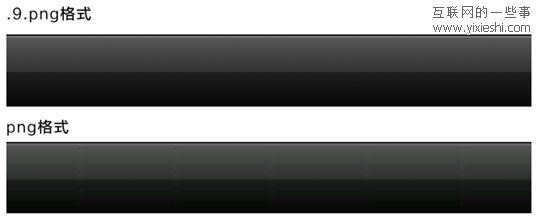
2. 你的.9.png必須繪有拉伸區域的黑線; 一些事

否則，圖片不會通過android系統編譯，導致程序報錯。還有，有同學疑惑解壓縮apk文件後，.9.png圖片里的黑線怎麼沒了? 一些事

那是因為andriod程序在把文件打包成apk的時候，程序會自動把\*.9.png圖片邊緣的黑線去掉，所以解壓縮apk後看到的.9.png文件是沒有黑線的。

三.使用「點九」的意義 網際網路的一些事

關於下圖，經過測試發現使用普通png的顯示效果出現明顯的變色橫紋。而.9.png圖片的顯示效果明顯優於普通png。



使用.9.png格式後，橫紋問題基本已解決。因為對於.9.png圖片，android系統程序有對其優化的算法。

由於android手機螢幕的材質質量差距大。很多螢幕不支持16位以上的顏色顯示。

所以渲染後結果出現丟失顏色，故造成橫紋顯示。

經與多款android手機對比後發現，螢幕越次的手機橫紋越明顯。

而使用了\*.9.png圖片技術後，只需要採用一套介面切圖去適配不同的解析度，而且大幅減少安裝包的大小。而且這樣程序不需要專門做處理的就可以實現其拉伸，也減少了代碼量和開發工作量。

相信每個人android平台的切圖工作，會有不同的技巧和心得，非常歡迎大家能在評論中留下自己的個人經驗及心得。相互交流會讓我們的工作模式更靈活，更高效，同時帶給大家更多優質的移動端應用：) 網際網路的一些事

Android.9圖片講解

2014/11/13 來源：CSDN博客

至於什麼是.9圖片這裡就簡單提一下，即圖片後綴名前有.9的圖片，如pic.9.png、pic1.9.jgp，諸如此類的圖片就稱為.9圖片。

①.9圖片的作用是在圖片拉伸的時候特定的區域不會發生圖片失真，至於哪些部分不會失真那麼得看你的圖片是如何繪製的。

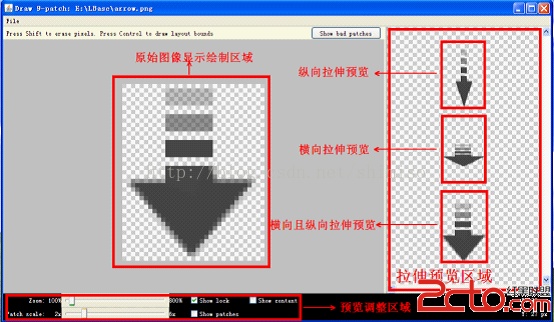
②當一張圖片作為背景時可以指定前景內容的顯示區域。如圖：



其中灰色是背景、綠色是前景。那麼前景放在背景的具體位置是可以通過.9圖片進行規定的。

①利用android開發環境中提供的工具draw9patch。

draw9patch如圖 ：



【draw9patch.bat其他功能說明】

Show lock：顯示不可繪區域

Show patches：預覽這個繪圖區中的可延伸宮格(粉紅色代表一個可延伸區域)

Show patches：預覽視圖中的高亮區域(紫色區域) 一些事

Show bad patches：在宮格區域四周增加一個紅色邊界，這可能會在圖像被延伸時產生人工痕跡。如果你消除所有的壞宮格，延伸視圖的視覺一致性將得到維護。

在「原始圖像顯示繪製區域」（這個名稱自己取的）其實顯示的是一張原始圖片，並且在原始圖片的四周添加了一個像素，通俗點說就是圖片比原始圖片擴大了一圈，多出來的那一圈是透明像素。我們需要繪製的地方真是在那一圈透明像素上進行繪製。在繪製之前先講講那一圈透明像素的含義。如圖：  
  


四周分別命名為L、T、R、B。

繪製在L的區域：用於拉伸的縱向區域。

繪製在T的區域：用於拉伸的橫向區域。

繪製在R的區域：用於顯示前景的縱向範圍。

繪製在B的區域：用於顯示前景的橫向範圍。

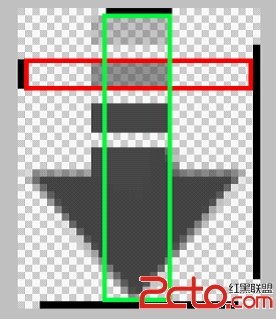
舉例說明一下，如圖 ：



這個圖會如何拉伸、如何顯示前景呢？最簡單的方法是直接看「拉伸預覽區」，當然更快捷是方法是你能直接看出這個區域。

拉伸區域

如圖：



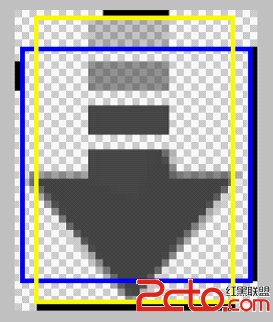
紅色框區域：表示縱向拉伸的區域，也就是說，當圖片需要縱向拉伸的時候它會只指定拉伸紅色區域，其他區域在縱向是不會拉伸的。

綠色框區域：表示橫向拉伸的區域，也就是說，當圖片需要橫向拉伸的時候它會只指定拉伸綠色區域，其他區域在橫向是不會拉伸的。

顯然紅色和綠色相交的部分是既會進行橫向拉伸也會進行縱向拉伸的。

前景的顯示區域

如圖：



藍色區域：表示前景能顯示的縱向範圍。即前景的最上面可以顯示到什麼地方，最下面可以顯示的什麼地方。

黃色區域：表示前景能顯示的橫向範圍。即前景的最左邊可以顯示到什麼地方，最右邊可以顯示的什麼地方。

藍色和黃色相交部分：表示整個前景能顯示的區域。一個區域是矩形的，藍色規定了上下邊界，黃色規定了左右邊界，兩者共同當然也就規定了一個矩形區域。

②利用photoshop進行繪製。

如何繪製？

A、首先將畫布四周放大一個像素。

B、然後在四周繪製黑色像素（四周像素的意義參見①）。

C、保存的時候改成.9圖片的後綴名。

何時使用ps繪製？

A、圖片很大的時候用draw9patch不是很好繪製，這個時候你就可以利用ps進行繪製。

B、當你要做很多圖標的時候，你可以利用ps中的動作功能來批量完成。

小結：如果你熟悉ps的話某種程度上可以做到事半功倍。而且可以做一些模板，或者是錄製一些動作。

簡單點講，一張圖片被拉伸以後會發生嚴重的圖像失真時，那麼就可以考慮.9圖片。那麼哪些時候這種圖片格式是比較有效的呢？

①圖片有圓角並且其他某一區域可以拉伸而不影響圖片效果。

②一些圖標需要適配不同的解析度的設備。

當然具體的使用還得看具體的要求，這裡只是說了一下我最常用的，不常用的就沒有列舉了。

PS：.9圖片很多時候你知道怎麼使用就可以了，至於為什麼會產生拉伸，類似這樣的問題你最好去問問google。

有兩點需要特別注意下：

1.最外圍的一圈像素必須要麼是純黑色，要麼是透明，一點點的半透明的像素都不可以有，比如說99%的黑色或者是1%的投影都不可以有;

2.文件的後綴名必須是.9.png，不能是.png或者是.9.png.png，這樣的命名都會導致編譯失敗。

Andorid中.9.png圖片的使用及製作

2016/01/05 來源：CSDN博客

我們有一個TextView，其裡面的內容是可以通過代碼動態改變的，我們想用一張圖片作為TextView的背景，實現類似於手機QQ對話中的氣泡文本效果。

TextView定義如下：

<TextView android:text="@string/hello\_world"android:layout\_width="wrap\_content"android:layout\_height="wrap\_content"android:gravity="center"android:background="@drawable/original"/>

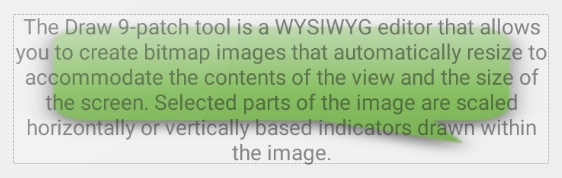
背景圖片如下所示：

最終TextView的效果如下所示：



上圖中的虛線是TextView的外輪廓，我們把TextView的寬度和高度都定義為wrap\_content，並且將gravity設置為center，但是還是沒有實現我們理想的狀態。TextView中的文本相對於整個圖片居中了，但是我們想要的效果是文本在綠色的矩形區域內居中。

上圖中的文本較短，展現出的問題還不是很明顯。當我們將TextView中的文本設置為具有一定長度的時候，問題更加突出，如下所示：



上面這張圖片暴露出兩個問題：

1. 我們原始的圖片是自上而下綠色逐漸變淡，並且右側和下側有陰影。當背景圖被拉伸為上圖大小的時候，圖片失真嚴重：消息框的四個角模糊不清；陰影被模糊放大，比較難看；綠色不清晰，漸變效果也基本看不出來。

2. 文字已經超出了圖片的消息框範圍，雜亂無章。

出現上述問題的原因是，整個背景圖被完全拉伸了，而且拉伸後的圖片的寬高比例與原始圖片的寬高比例不同。如果只對原始圖片矩形消息框中的區域進行拉伸，那麼就可以解決上述問題。

為了解決上述問題，我們可以使用Android中的.9.png格式圖片，即Nine-patch。

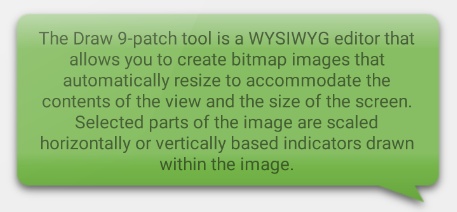
下面對.9.png格式圖片進行一下說明：

1. 首先，.9.png格式的圖片本身就是.png格式圖片，不過該圖片以.9作為文件名的後綴。

2. 我們可以將一個.png圖片轉換成.9.png圖片，.9.png圖片比正常的.png圖片在圖片最外圍的四邊多了1px邊框，我們可以在這最外層的1px邊框上定義圖片的可拉伸區域以及圖片的內容區域。

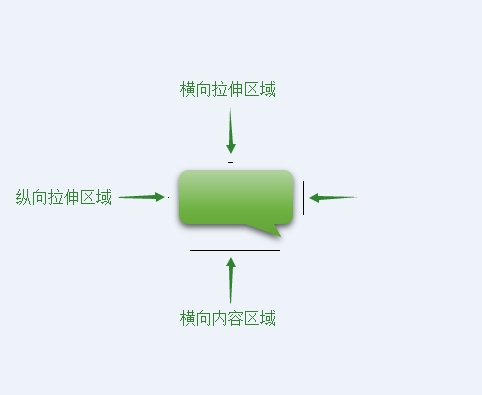
我們對原始的背景圖進行編輯，可以得到如下的文件名為message.9.png的圖片：

當我們用該處理過的.9.png圖片作為TextView的背景時，效果如下所示：



由上圖我們可以看到，所有文本都居中放到了消息框的矩形區域中，而且圖片沒有失真，保持了原有圖片的質感。

我們可以仔細觀察一下上述的message.9.png這張圖片，可以發現該圖片的最外層的四邊有黑點、黑線，具體來說：



1. 可以將圖片最上側1px邊框中的一個或多個點設置為黑色，本例中我們用了一條黑色的線段，這些黑色的點定義了圖片中可以被橫向拉伸的區域。 同樣也可以將圖片最左側1px邊框中的一個或多個點設置為黑色，本例中我們只用了一個黑色的像素點，這些黑色的點定義了圖片中可以被縱向拉伸的區域。橫向拉伸像素點與縱向拉伸像素點相交定義了圖片中可拉伸的矩形區域，這樣就實現了對圖片中一部分區域進行拉伸。

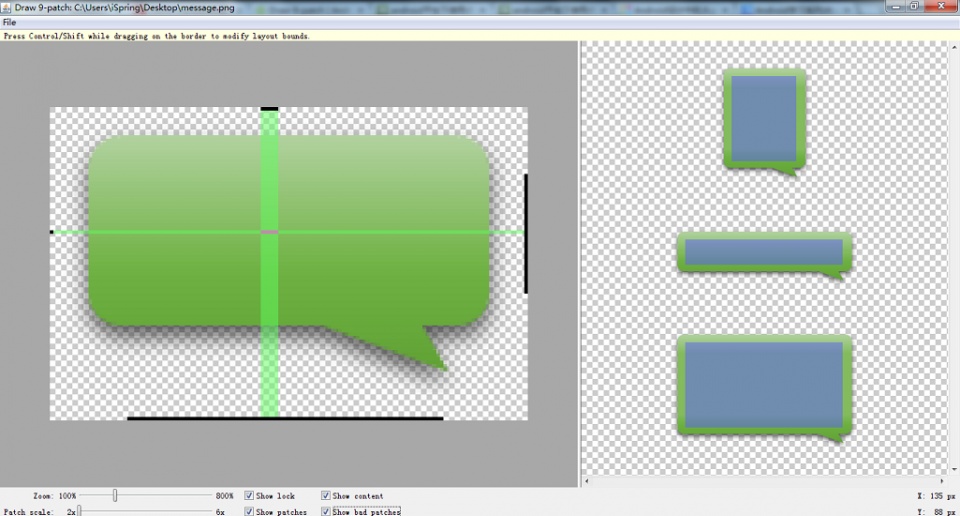
2. 我們可以選擇性地對圖片的底邊和右邊設置黑色線段，用這些黑色線段定義圖片的內容區域。當我們圖片里的樣式是不規則圖形的時候，定義圖片的內容區域很重要，TextView中的文本都會放到內容區域中。將圖片最下側1px邊框設置一條黑色線段，該橫向線段定義了圖片的橫向內容區域。將圖片最右側1px邊框 設置一條黑色線段，該縱向線段定義了圖片的縱向內容區域。橫向線段與縱向線段的組成的矩形區域就組成了內容區域。如果不定義圖片的內容區域，那麼圖片的內容區域就是整個圖片區域。

3. .9.png最外側四邊中的像素要麼是純透明、純白色，要麼是純黑色，不要設置其他顏色和透明度。

我們可以用Photoshop將一個已有的.png編輯成.9.png。其實，Android本身也提供了編輯.9.png的工具，Android/sdk/tools目錄下有很多實用的工具，我們可以使用該目錄下的draw9patch工具對.png圖片進行編輯。

可以通過命令行，也可以通過直接雙擊啟動draw9patch，顯示出GUI介面。

打開「Draw 9-patch」之後，可以直接將一個.png圖片拖拽到GUI介面上，也可以通過File菜單中的「Open 9-patch」菜單項打開.png圖片。左側是圖片編輯區，右側是預覽區。可以在編輯區對圖片的最外層的四邊框設置黑色點，如下圖所示：



通過滑鼠單擊可以將最外層中的像素設置為黑色，通過這種方法可以設置圖片的拉伸區域和圖片的內容區域。按住Shift鍵再單擊黑色像素可以將黑色像素重置為透明。左側的編輯會實時在右側區域顯示出來。右側預覽區中有三個圖片，第一個圖片表示的是垂直方向進行拉伸的預覽效果圖，第二個圖片表示的是水平方向進行拉伸的預覽效果圖，第三個圖片表示的是同時在水平和垂直方向上進行拉伸的預覽效果圖。

在介面下側是控制參數的面板。

Zoom

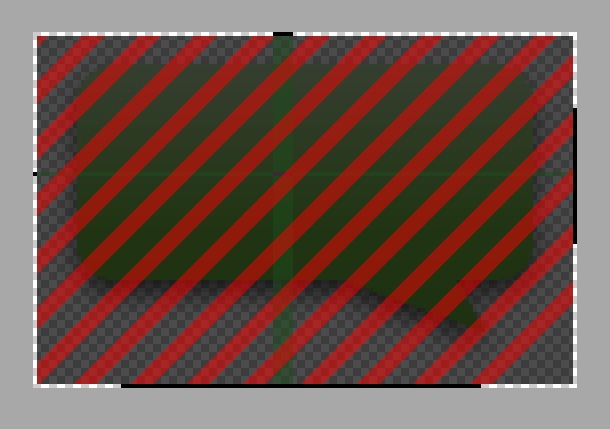
通過調節Zoom可以調節左側編輯區域的縮放比例。

Patch scale

通過調節Patch scale可以調節右側預覽區域的縮放比例。

Show lock

我們之前提到了，我們只能編輯圖片最外層1px的邊框，如果勾選了Show lock，那麼當我們將滑鼠放到左側編輯區域時，那麼不可編輯區域會以紅色條紋遮蓋，效果如下所示：



Show content

當我們勾選了Show content後，右側預覽區中的圖片會把內容區域以藍色表示。

Show patches

當勾選了Show patches後，左側編輯區域中的可拉伸區域會粉紅色高亮顯示出來。

Show bad patches

當我們在top或left邊上設置可拉伸區域設置了連續多個像素時，有可能會造成圖片在拉伸時多個像素顏色進行插值導致失真，如果出現了這種情況會以紅色顯示。

希望本文對大家使用.9.png有所幫助！

超詳細解說點9切圖的製作方法（內附神器）

2016/09/20 來源：莫貝網

點九圖，是Android開發中用到的一種特殊格式的圖片，文件名以」.9.png「結尾。這種圖片能告訴程序，圖像哪一部分可以被拉升，哪一部分不能被拉升需要保持原有比列。運用點九圖可以保證圖片在不模糊變形的前提下做到自適應。點九圖常用於對話框背景圖片中。

之前，點9圖的出現是安卓機為了適配不同的手機解析度的解決方案。

點9圖（.9.png）是用於Android開發的一種特殊的圖片格式，它的好處在於可以用簡單的方式把一張圖片中哪些區域可以拉伸，哪些區域不可以拉伸設定好，同時可以把顯示內容區域的位置標示清楚。現在點9圖不僅可以應用在安卓上，同樣可以應用到ios切圖上。

唯一的區別：安卓的.9.png必須用draw9patch.bat（蘋果電腦下載：draw9patch.jar）這個工具來製作。我們平常用photoshop直接輸出的.9.png只是普通的png資源。但是ios是可以使用的。安卓工程師是不能用的。 或者說如果你的9.png中沒有黑線，安卓開發軟體是不識別的。

嚴謹的來說：APP設計師或安卓工程師必須用自帶的draw9patch.bat工具轉換下。



draw9patch.bat工具的下載地址：立即下載

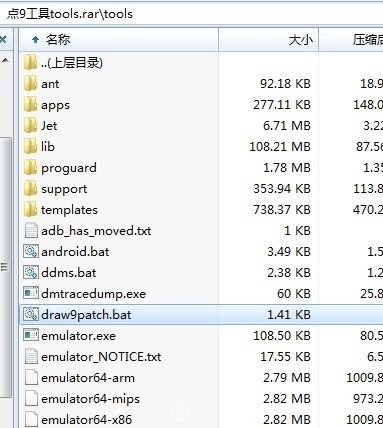


win

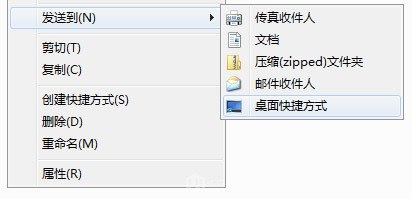


mac

下載後解壓，打開裡面的draw9patch.bat



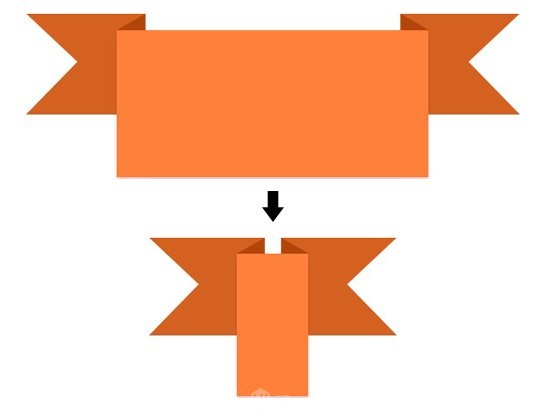
最好是添加一個快捷鍵在桌面



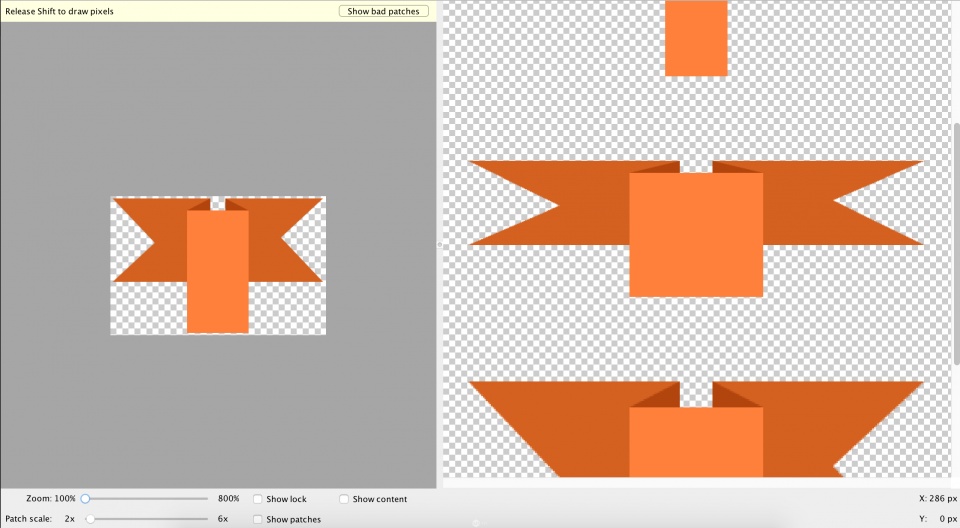
蘋果直接打開就可以用了

下面我以絲帶為例，講講具體的製作方法：

1.打開draw9patch.jar，把導出的PNG文件拉進去（圖片四周要預留1px）

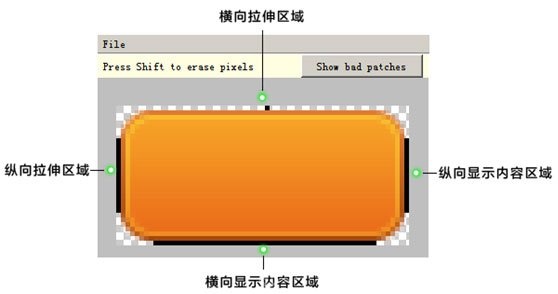


ps：在處理圖片的時候，你可以先縮短想要拉伸的部分，



2.我們在圖片邊緣點擊左鍵，繪製出黑線，即圖片需要被拉伸的部分。

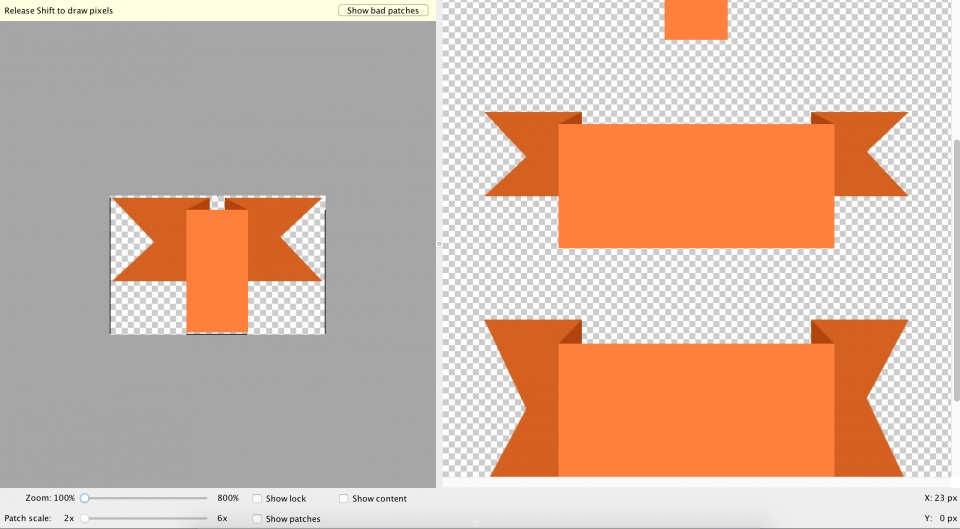
下圖，先給大家上一張圖讓大家了解4條黑線是什麼意思



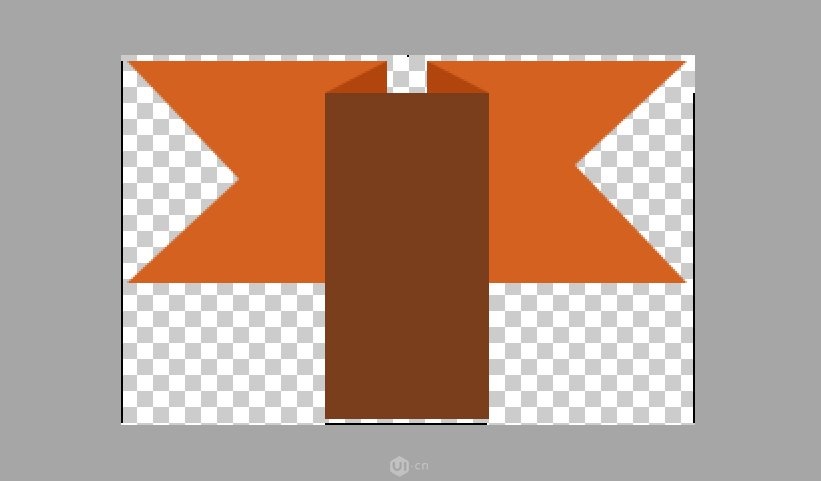
點九圖格式規定由左側和上側來控制圖片的拉伸，右側和下冊控制文字的顯示區域。是什麼意思呢？

拉伸區域：當實際程序中設定了對話框的寬高時，橫向拉伸區域和縱向拉伸區域就會被拉伸成所需要的高和寬，呈現出於設計稿一樣的視覺效果。

內容區域：為了讓文字被輸入框完整包裹，以達到顯示效果的正常，所畫黑線的區域就是你文字顯示的區域，這樣你的文字就不會超出對話框。(如果你害怕出錯也可以把整一邊都畫上，讓研發去調試,或者事先找研發問一下)



這是文字就會被控制在這一塊(棕色)以內



如果失誤多繪的部分，可按住shift鍵的同時點擊滑鼠左鍵擦除。

【draw9patch.bat其他功能說明】

Zoom: 用來縮放左邊編輯區域的大小

Patch scale: 用來縮放右邊預覽區域的大小

Show lock：當滑鼠在圖片區域的時候顯示不可編輯區域

Show patches：在編輯區域顯示圖片拉伸的區域 （使用粉紅色來標示）

Show content：在預覽區域顯示圖片的內容區域（使用淺紫色來標示）

3.畫好以後右邊先看看效果，如果都沒問題，導出，可以發給程序猿了，搞定。

Android開發 -- .9圖片原理和Draw 9-patch工具的使用

2015/05/13 來源：CSDN博客

一. .9圖片的介紹

.9圖片的作用：

1. 主要防止圖片被拉伸後變形。因為在一個像素里如果是相同色塊，那麼它被拉伸不會變形。但如何該像素里有不同色塊，就將導致它變形。

2.通過.9圖片可以把圖片做的很小，不占資源，減少apk的大小

下面我們來看一下，同一張圖片使用了.9和不使用.9圖片的區別：



由圖中，我們可以看出沒有使用.9圖片的對話框被拉伸的變形了。

我們再看下這倆個背景圖片的區別：(左：.9圖片 ； 右：非.9圖片)

仔細觀察，才發現.9圖片在左邊和上面多了一個小黑點，其他基本一致。

二. .9圖片的使用1 -- 拉伸圖片

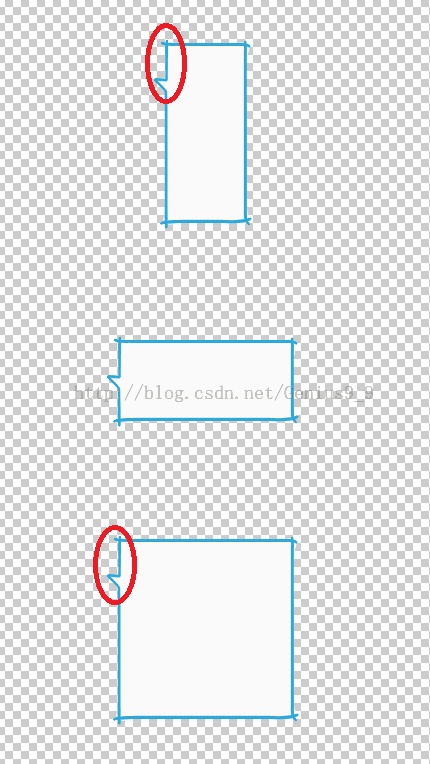
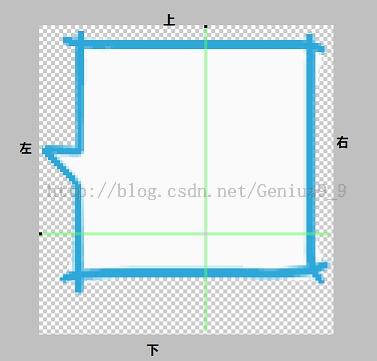
（關於圖片下方和右方的作用在下一個實例中會講解）

在Draw 9-patch工具中，選中**show patches**會自動顯示出被拉伸的區域。

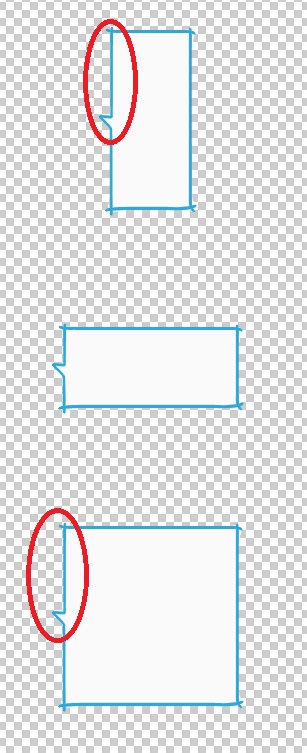
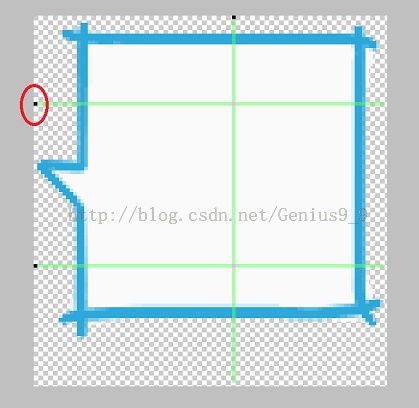
在Draw 9-patch工具的右手邊，可以預覽拉伸效果，從上到下依次是：縱向拉伸 / 橫向拉伸 / 縱向橫向拉伸

下面我們來看一下倆種不同的拉伸區域的.9圖片造成的不同效果：

a.



b.



圖a和圖b的不同點在於： 圖a的左側只在箭頭的下面畫了個黑點，而圖b的左側在箭頭的上下面都各自畫了個黑點。

從右手邊的預覽圖，我們也可以看出圖a在縱向拉伸時，只會拉伸箭頭下方那個黑點所在的區域。而圖b在縱向拉伸時，箭頭上下方的區域都會進行拉伸。

為了更直觀的看出效果，可在Draw 9-patch工具中，**Patch scale**是進行拉伸的倍數選擇，可將它由最小值滑到最大值，以在預覽圖中觀察效果。

同時，我們在真機中進行編譯看效果：



三. .9圖片的使用2 -- padding效果

1.下：黑色區域表示橫向的**內容**顯示區域

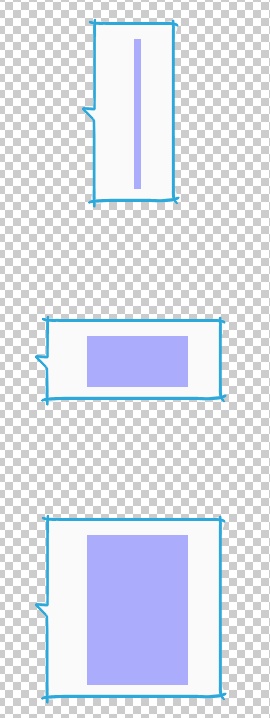
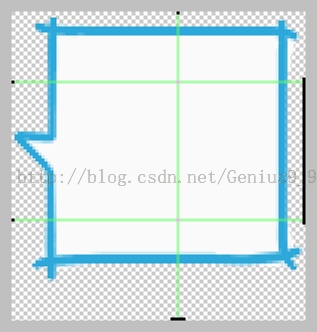
2.右：黑色區域表示縱向的**內容**顯示區域

在Draw 9-patch工具中，勾選**Show content，**可以在右手邊的預覽區域看到紫色的區域就是內容顯示的區域。

我們來看一個使用了下方和右方黑點的和沒有使用的區別：



由圖中，我們可以很直觀的看出，使用了下方/右方黑點的可以適當調整文字內容在圖片中的區域。其實就是實現android:padding的效果，即設置內邊距。



下方黑點區域的左端實現 android:paddingLeft ，右端實現android:paddingRight

右方黑點區域的上端實現android:paddingTop，下端實現android:paddingBottom

同樣，可在Draw 9-patch工具中，**Patch scale**是進行拉伸的倍數選擇，可將它由最小值滑到最大值，以在預覽圖中觀察效果。

還不明白點九圖？還沒看這篇文章吧！

**作者**：菜心

很多朋友說看了網上很多關於點九的文章，越看越糊塗，之前也有答應大家，科普一下點九圖，所以今天文章來啦！

先來看看大綱：

**1. 什麼是點九圖**

**2.點九圖的作用是什麼**

**3.點九圖的原理**

**4.總結**

**1. 什麼是點九圖**

點九圖其實就是安卓系統中特有的一種圖片格式，為了讓大家更好的記住，我們只要知道，後綴名是「.9.png"的圖片，就是點九圖。

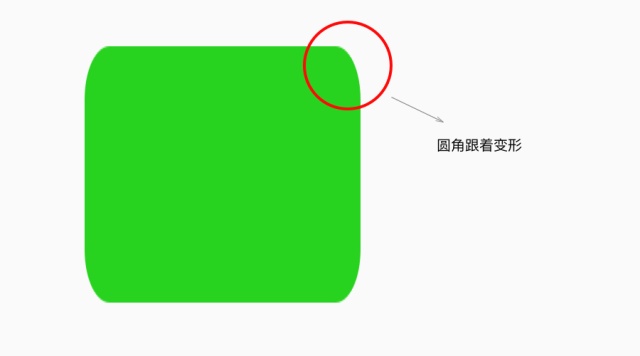
**2.點九圖的作用是什麼**

每個事物都有其存在的價值，所以先弄清楚點九圖能為我們解決什麼問題，這樣後面理解起來就會容易很多。

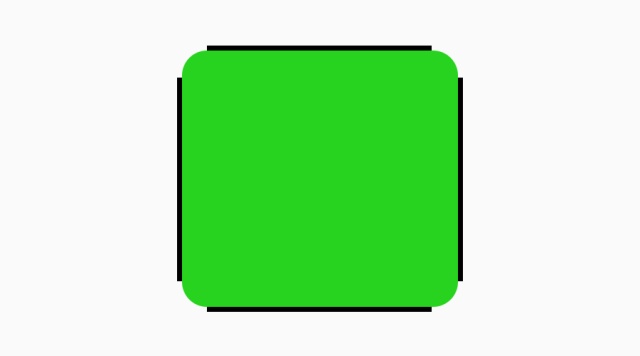
其實點九圖的用處就是幫助我們拉伸切圖的，比如這個切圖：



如果它需要縱向拉伸，直接拉會變成下面這個樣子：



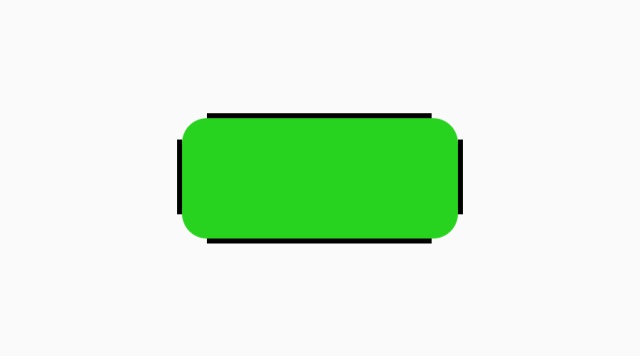
而用了點九圖就可以讓切圖局部拉伸，而不是整體拉伸，這樣就可以把容易變形的地方保護起來：



是不是拉伸的很完美！有黑線不用怕，那只是告訴安卓系統：「嘿，大兄弟，這是張點九圖，特殊對待一下！」

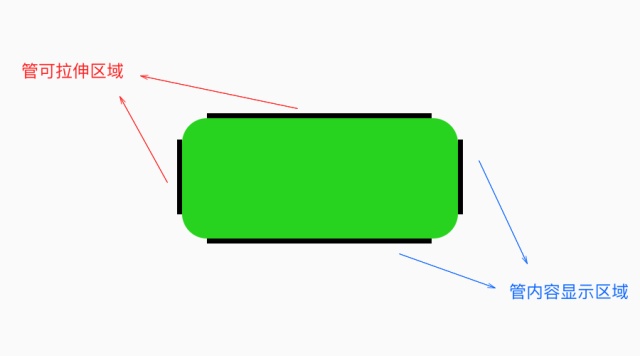
**3.點九圖的原理**

點九圖最大的原則就是四個邊必須各有一條純黑的線（或一像素的點），如下圖：

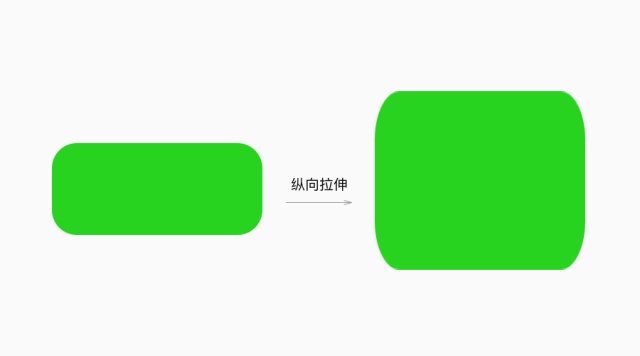


如果四條線少任何一條，或者線不是純黑的（#000000），安卓系統就不認你！

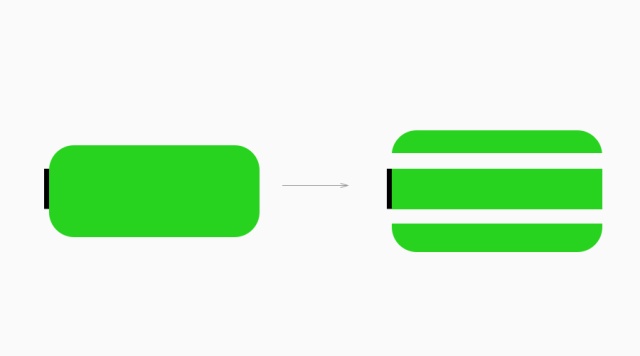
其中這四條線，左上兩條線掌管可拉伸區域，右下兩條線掌管內容顯示區：



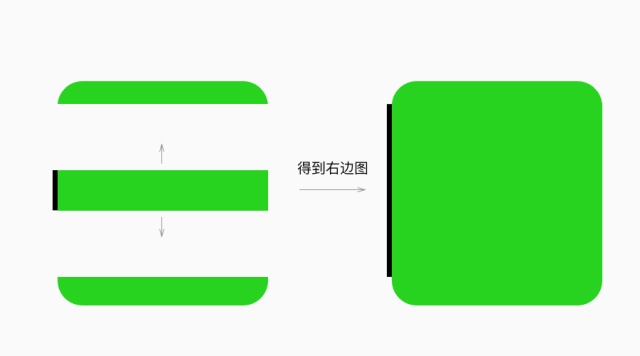
先說左上兩條線，因為兩條線原理是一樣的，所以我們單獨拿左邊這條線來舉例吧，當我們沒有左邊那條黑線時，縱向拉伸是這樣的：



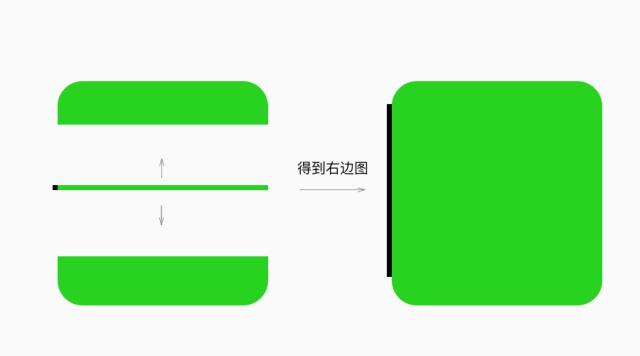
圓角是不是變形了，如果左邊加一條黑線，就相當於把原來的圖形分為三個部分：



當再次縱向拉伸的時候，只有標記了黑線的部分可以被拉伸，而上下兩部分是完好無損的，想像一下，無論你拉伸到多高，是不是都不會變形了：



當然，你左邊畫一個點也可以起到相同的效果：

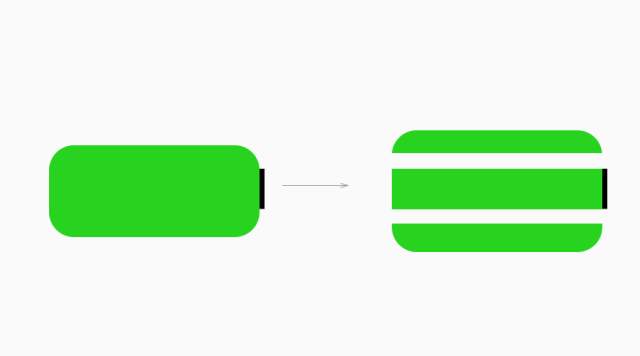


左邊這條線是控制縱向拉伸的，所以上面那條黑線就是控制橫向拉伸的，原理是一樣的！

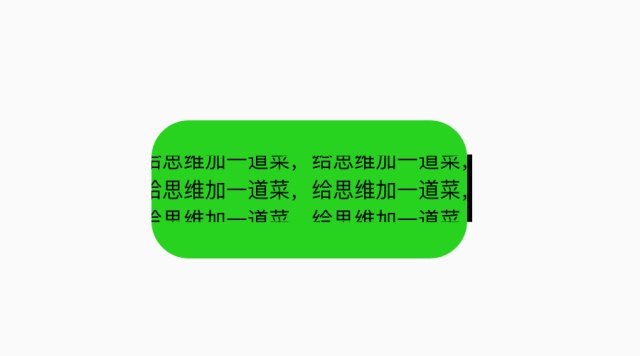
接下來我們來說右下兩條線，是控制內容區域的，單獨拿右邊那條線來舉例，如果沒有右邊那條黑線，在這個切圖上輸入內容，比如文字，是沒有限制的，內容會撐滿整個背景圖：



當有了右邊那條黑線後，切圖相當於在縱向上又被分開：

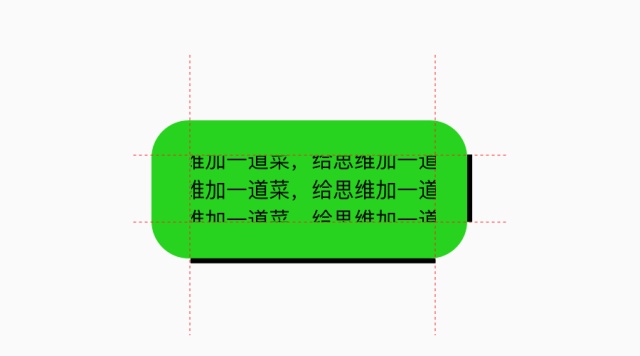


而這次是右邊有黑線哦，別忘了右邊的黑線是控制顯示區域的，所以只有帶黑線的部分才可以顯示內容



當然，文字不會這樣被切斷一半顯示的，這裡只是方便大家觀看哪裡可以顯示內容！

同樣的原理，當我下面也畫一條線後，橫向上也是只有帶黑線的部分可以顯示內容（紅線是輔助示意的哈）：



這就是點九的基本原理了！

**4.總結**

最後總結幾個要點：

1.點九切圖四周必須要有四條一像素純黑的線或點

2.左上兩條線控制拉伸區，右下兩條線控制內容區

3.輸出的圖片後綴必須是「.9.png"

至於用插件還是自己手動切點九圖，看自己習慣吧，我一般都是自己畫，比較放心一點。

好了，今天就講這些了，我猜你應該大概明白了，如果沒明白，再看一遍吧。

科科！