**Lab3**

**物件控制與監聽事件**

本節目的：

* 透過程式碼使用Xml畫面元件。
* 設定監聽事件回應使用者操作。
  1. 元件與監聽事件
     1. 取得畫面元件

前面章節我們完成了Layout（Xml）的設計，而實際要去使用這些元件，我們需要把畫面中的元件與主程式（Java）連結。一開始，我們會在Activity來編寫我們的程式碼。

在建立好的專案中，會配置一個程式檔：MainActivity以及一個布局配置：activity\_main.xml，如圖3-1所示。

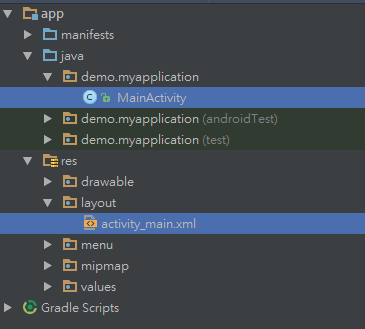


圖3-1 Android專案架構

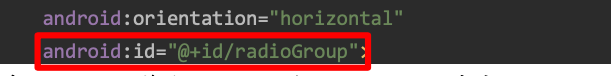
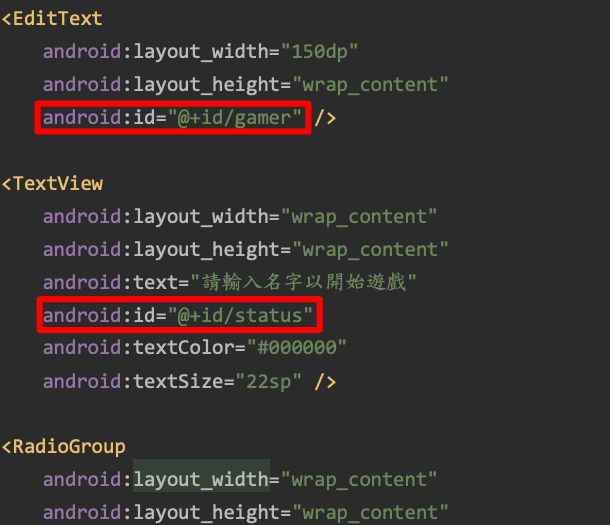
首先activity\_main.xml延續使用第二章節的畫面設計，配置結果如圖3-2所示下。

|  |  |
| --- | --- |
|  |  |

圖3-2 猜拳遊戲預覽畫面（左）與布局元件樹（右）

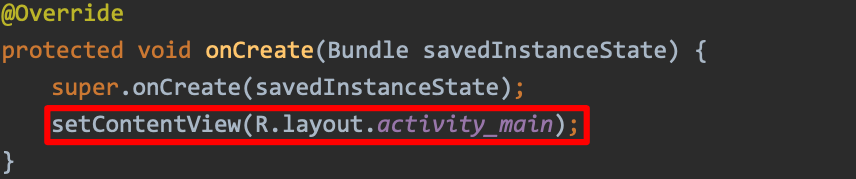
圖3-2中的元件如果希望能在程式中使用，則在xml上需要設定一組對應的id，id的用途是為該元件貼上一個識別標籤，在Android中必須透過id才能找到對應的元件。

如下方的程式碼片段中，EditText的id為gamer、TextView的id為status、RadioGroup的id為radioGroup…以此類推。



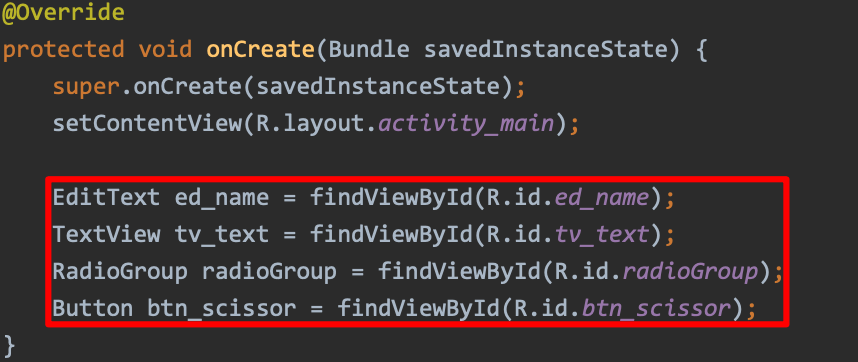
完成了畫面設計，接著我們便可以在MainActivity中去取得activity\_main.xml的元件。

打開MainActivity我們可以看到以下程式碼：

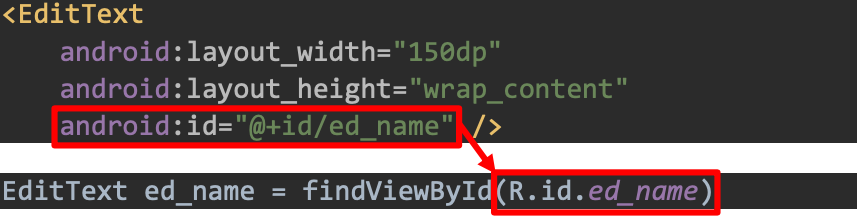


由於Android主程式與Xml畫面本身是不同的程式碼，如果要在主程式中要顯示畫面，就必須要明確指定要使用哪頁Xml畫面。setContentView()方法會指定這個Activity所要使用的Xml畫面，這邊系統已經事先指定了（R.layout.activity\_main），也就是透過R類別指定了activity\_main.xml。

在Java中，當我們要取得Xml內的元件時，必須是使用findViewById()將元件綁定，如以下程式碼所示：



findViewById()方法會從Xml中找到對應id的元件（R.id.xxx），並回傳元件到程式碼，如上面xml程式碼的ed\_name、tv\_text、radioGroup、btn\_scissor等。



findViewById()回傳的結果為View型態，View型態是畫面元件的原型類別，所有畫面元件都屬於View型態，如EditText、TextView、Button，如圖3-3所示。

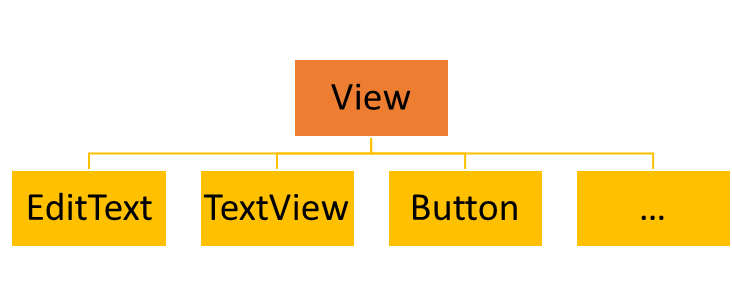


圖3-3 View成員

回傳View型態，也就表示findViewById()只能知道他取到了是一個畫面元件，但是不知道是什麼元件，因此只能回傳說找到了一個「畫面元件」。所以在程式碼當中，我們需要明確的告知這是EditText元件，主要就是將findViewById()回傳的View型態，並藉由<EditText>的型態語法，轉型成EditText元件。

編寫完findViewById()後，應該會發現EditText、TextView、Button都是圖3-4的EditText會顯示**Unresolved…**，因為要使用這些元件時，程式碼中還必須匯入對應的android的widget類別，我們必須在程式碼上加入Import。



圖3-4 找不到元件的class

Import可以手動加入，也可以讓Android Studio自動產生，用鼠標點擊有問題的元件，如圖3-5所示。然後按下Alt+Enter後就可以自動匯入，如圖3-6所示，或是會彈出一個小列表，這時選擇需要的class自動匯入。



圖3-5 匯入需要的class



圖3-6 Import EditText元件後就不再顯示錯誤

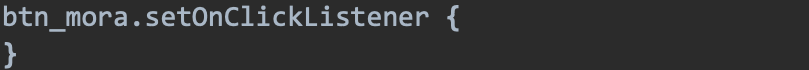
* + 1. 事件處理

使用者在操作APP的過程中，會對於畫面物件產生事件，例如點擊、輸入、觸碰等，而程式中透過對物件設置監聽器去等待事件被觸發。如點擊某個元件，程式碼就會被啟動去做特定的工作，這種回饋動作就是透過監聽器來實現。

Android有提供內建的監聽器給元件做使用。不同類別的元件，能使用的監聽器也不同。而監聽器的命名規則通常為On\_XXX\_Listener，如OnClickListener，每個監聽器內部會預定幾種方法，當觸發時會執行對應的方法。

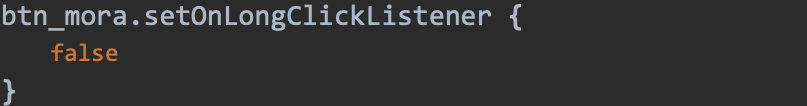
而監聽器種類繁多，以下則會就幾個常見的監聽器做說明：

* OnClickListener：



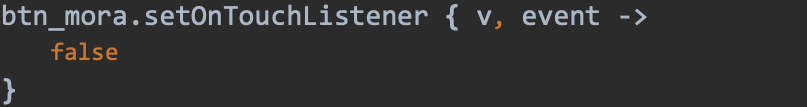
OnClickListener是最常使用的監聽器，他可以在使用者對元件做出**點擊（按下後立刻放開）做出回應**。該元件使用setOnClickListener()的方法將該監聽器做連結，之後就可以等待該元件觸發按下的動作。

* OnLongClickListener：



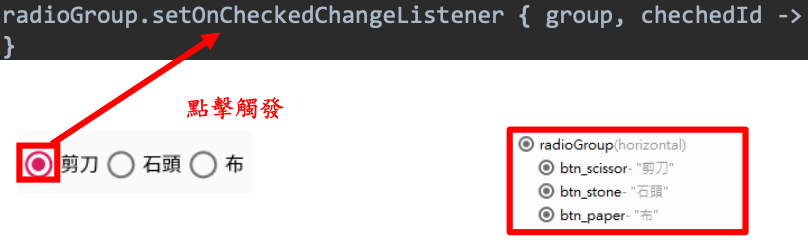
OnLongClickListener可以在使用者對元件做出**長按（持續按住不放開超過1秒）做出回應**，當事件成立時會觸發，元件使用setOnLongClickListener ()的方法將該監聽器做連結。

* OnTouchListener：



OnTouchListener提供使用者對元件做出的**觸摸事件（按下、移動手勢、離開…等）做出回應**，當事件成立時會觸發setOnTouchListener，參數event為觸發的事件。

* OnCheckedChangeListener：(RadioButton必須是RadioGroup的子層級)



OnCheckedChangeListener是RadioGroup專用的監聽事件，它的用途在於監聽RadioGroup子層級的RadioButton被按下時，會觸發onCheckedChanged()。

onCheckedChanged()的第二個參數，會回傳剛才按下的元件Id，或是我們也可以用radioGroup.getCheckedRadioButtonId()的方法取得元件Id。

這邊補充說明，在選擇監聽器時，Android Studio會在我們打字的同時用表單列出可用的語句，要設定監聽器時可以打上setOn…就可以篩選出下圖3-7中能用的監聽事件。

說明



圖3-7 加入按鈕監聽事件

* 1. 猜拳遊戲程式設計
* 延續前一個Lab的結果，實做一個完整的猜拳功能，如圖3-8所示。

|  |  |
| --- | --- |
|  |  |

圖3-8 猜拳遊戲實機畫面

* 選擇剪刀石頭布三個選擇之一，並按下開始遊戲，系統會以亂數回應勝負，下方會顯示出輸入的名稱、勝利者與雙方所出的拳，如圖3-9所示。

|  |  |  |
| --- | --- | --- |
|  |  |  |

圖3-9 玩家勝利（左）、平手（中）、電腦勝利（右）

* + 1. 加入監聽與判斷式

開啟MainActivity，並對Button設定OnClickListener的監聽事件，在這裡我們加入姓名的判斷與剪刀石頭布的判斷式。





TextView可透過程式碼直接修改text屬性，而EditText必須使用setText()才能設定字串內容。為了確保資料類型一定為字串，從TextView與EditText取得text屬性時，建議要再額外加上toString()的轉型取得字串型態的資料。

說明