

Chapter

07

多媒體

Camera 元件主要功能是啟動行動裝置的照相機來照相。**ImagePicker** 元件功能會自動開啟行動裝置的相簿，讓使用者可以從相簿中選取一張相片。

Sound 元件可以播放聲音檔，其主要功能是播放較短的音效檔。**Player** 元件也是用來播放聲音檔，與 **Sound** 元件不同處，在於其主要是播放較長的音樂檔案。**SoundRecorder** 元件是用來錄製聲音檔。

Camcorder 元件是用來錄製影片檔。**VideoPlayer** 元件是用來播放影片檔，支援的影片格式有 **.wmv**、**.3gp** 及 **mp4**。



深入解析

Player 及 Sound 元件使用 Start 方法播放聲音檔時，會直接播出聲音，但是沒有控制面板，如要暫停、停止播放等功能，必須設計者自行安排。VideoPlayer 元件播放影片檔時，只要在影片播放區域點按一下，此時下方會出現影片控制面板，可控制影片播放，不需自行設計。



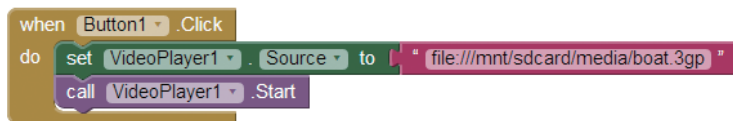
為了不影響畫面，影片控制面板在數秒後會自動消失，使用者任何時刻在影片播放區域點按一下，影片控制面板就會出現。

播放 SD 卡中的影片檔

影片檔的容量相當大，依解析度的不同，通常是聲音檔的數倍到數十倍，因此幾乎無法儲存在伺服器上，一般都是將影片檔放置在 SD Card 上。要播放 SD 卡上影片檔的路徑為：

```
file:///mnt/sdcard/SD 卡路徑
```

例如：按下 **Play** 按鈕播放 SD 卡中 <media> 資料夾的 <boat.3gp> 影片檔，其程式拼塊為：





7.3.3 整合範例：攝放影機

有了 Camcorder 及 VideoPlayer 元件，只要將兩者結合，就可建立既可攝影。又可觀看拍攝影片的應用程式。因為 VideoPlayer 元件提供了影片控制面板，程式比錄音機應用程式簡單許多。

▼ 範例：攝錄影機

程式執行時只有 **開始錄影** 鈕有作用，按 **開始錄影** 鈕會啟動行動裝置的攝影設備開始錄製影片檔，結束攝影時會回到應用程式，此後 **開始錄影** 及 **播放錄影** 鈕都有作用。按 **播放錄影** 鈕就會開始播放錄影檔，在影片播放區域點按一下，下方會出現影片控制面板。(<ch07\ex_Camcorder.aia>)



建議使用實機執行

本範例使用攝影功能，必須在行動裝置上執行。

» 介面配置



此處 VideoPlayer1 元件的 **Width** 屬性值設為「320」，**Height** 屬性值設為「240」，不論影片解析度為何，將在此範圍內播放。

» 程式拼塊

1. `buttonEnable` 程序設定兩個按鈕是否有作用：若參數值為 `true`，表示按鈕有作用，`false` 表示無作用。

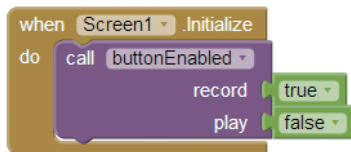
```

to buttonEnabled record play
do
  set ButtonRecord . Enabled to get record
  set ButtonPlay . Enabled to get play

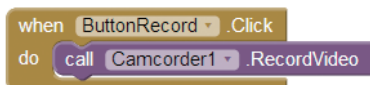
```



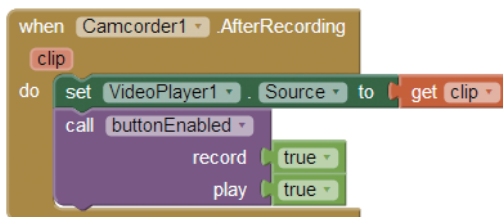
2. 程式開始時只有 **開始錄影** 鈕有作用。



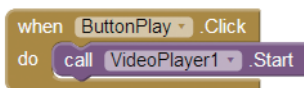
3. 按 **開始錄影** 鈕後就以 Camcorder 元件的 RecordVideo 方法開始拍攝。



4. 攝影完成會觸發 AfterRecording 事件並傳回影片檔路徑，程式會將影片檔路徑做為 MediaPlayer 的影片來源，並設定兩個按鈕都有作用。



5. 按 **播放錄影** 鈕後就以 MediaPlayer 元件播放影片。



Chapter

10

網際網路應用

WebView 元件主要用來顯示指定的網頁內容，它的功能等於在 **App Inventor 2** 中嵌入一個小瀏覽器，除了能夠顯示網頁的內容，也可顯示文字、圖片、**Gif** 動畫，甚至是 **Google Maps**。

Web 元件是屬於背景執行元件，執行時並不會顯示在 **Screen** 中，**Web** 元件可以將資料以 **Get**、**Post** 等方式傳遞到指定網址，再將資料讀取回來。



10.2 嵌入多功能的 Google Maps

WebView 元件除了用以製作網頁超連結和顯示文字，事實上，它還可以顯示圖片，甚至是 gif 動畫檔，此外，善用網址的語法，除了可以載入 Google Maps，還可以在地圖上建立多個地標，或是作路徑規劃。

Google 提供 Google Static Maps API 服務，產生出相關的靜態地圖，目前已不需要申請 Google API 金鑰，只需要瞭解相關參數設定 URL 網址，即可以做出靜態地圖。

靜態地圖參數

以「<http://maps.google.com/maps/api/staticmap?>」語法，並配合參數設定，可以建立靜態地圖。建立靜態地圖的語法：

```
http://maps.google.com/maps/api/staticmap? 參數一 & 參數二...& 參數 n
```

重要的參數如下：

參數	說明	
center	設定地圖中心點。	
zoom	設定縮放等級。	
size	圖片大小。(此參數必填)	
sensor	是否有感測器 (true 或 false)(此參數必填)。	
mapTypeId	設定地圖的顯示類型 (預設為 roadmap)。	
	名稱	顯示類型
	roadmap	顯示預設 2D 街景地圖。
	satellite	顯示衛星地圖。
	hybird	顯示衛星和街景混合的地圖。
	terrain	顯示具有高度、山嶽和河流的地圖。

例如：以「台北 101」為中心點、縮放等級為 15、圖片大小為 512x512，建立 2D 街景地圖及不使用感測器的靜態地圖語法如下：

```
http://maps.google.com/maps/api/staticmap?center=25.033194,121.56438&zoom=15&size=512x512&maptype=roadmap&sensor=false
```



建立地標

在地圖上建立地標，可以讓地圖有一個明確的標示點，具有畫龍點眼效果。
markers 可以在指定地點建立地標，並指定地標顏色和地標文字。

語法如下：

```
&markers=color: 地標顏色 | label: 地標文字 | 緯度, 經度
```

地標中 **color**、**label** 和 (緯度, 經度) 參數間，必須使用「|」隔開。

參數	說明
color	blue 藍色、green 綠色、red 紅色等。
label	設定文字 (0~9、A~Z)。

例如：以「台北 101」為中心點建立靜態地圖，並建立兩個地標，第一個地標為 A，顏色為藍色，第二個地標為 B，顏色為綠色。

```
http://maps.google.com/maps/api/staticmap?  
center=25.033194,121.56438&zoom=12&size=512x512  
&motype=roadmap&sensor=false  
&markers=color:blue|label:A|25.033194,121.56438  
&markers=color:green|label:B|25.0567,121.6071
```



第二個地標

第一個地標



建立 Label: 地標文字

建立的地標文字只可使用 **0~9**、英文字母 **A~Z** 的單一字元，否則將不會顯示地標文字，而只顯示地標圖示。

路徑規劃

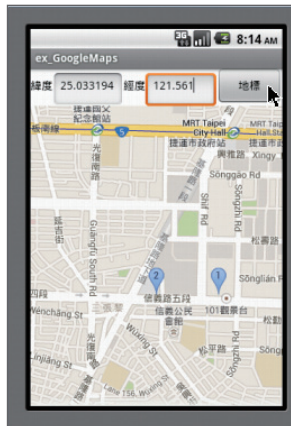
也可以使用 Google Maps 做路徑規劃，只要設定 **saddr** 為起點，**daddr** 為終點即可顯示規劃的路徑。

```
https://maps.google.com.tw/maps?&saddr=25.033194,121.564387  
&daddr=25.0567,121.6071
```

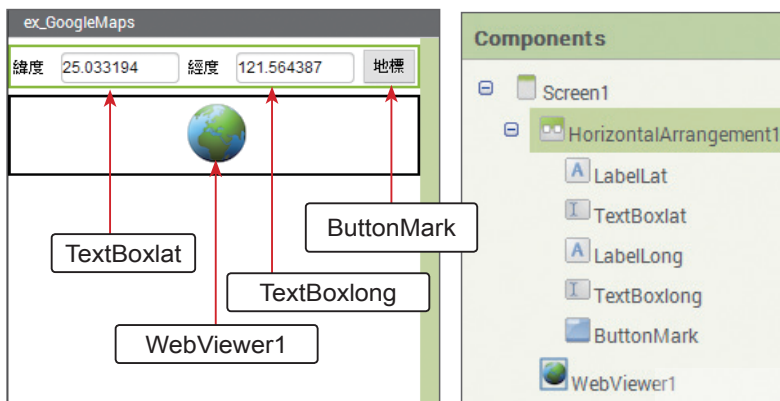



▼ 範例：指定地標的 Google Maps

在這個範例中我們將以 WebViewer 元件顯示 Google Maps，也可以輸入指定的緯度、經度後，按下 **地標** 按鈕建立地標。(<ch10\ex_GoogleMaps.aia>)



» 版面配置





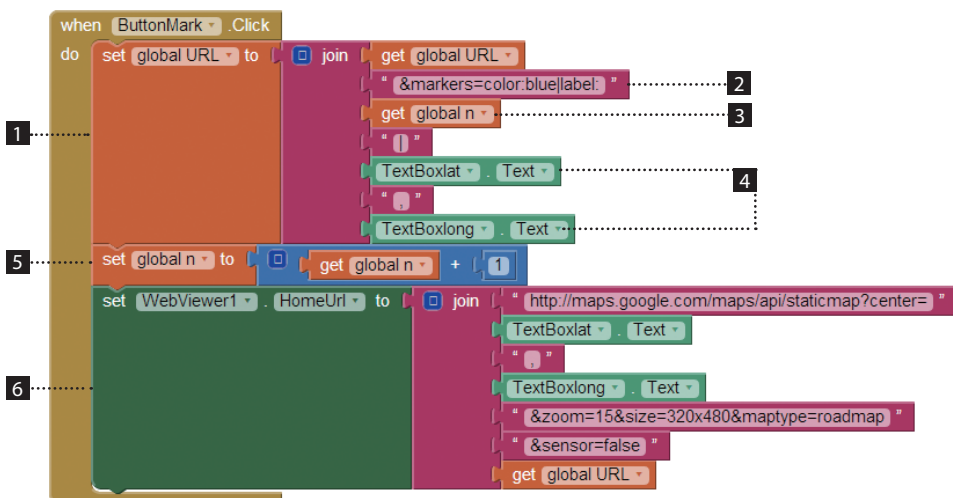
» 程式拼塊

1. 建立變數儲存經緯度和地標文字。



- 1 URL 儲存輸入的緯度、經度。
- 2 建立的地標文字，由 1 開始累加。

2. 按下 **地標** 鈕會以 TextBoxlat、TextBoxlong 的輸入值建立藍色地標，地標文字從 1 開始遞增。



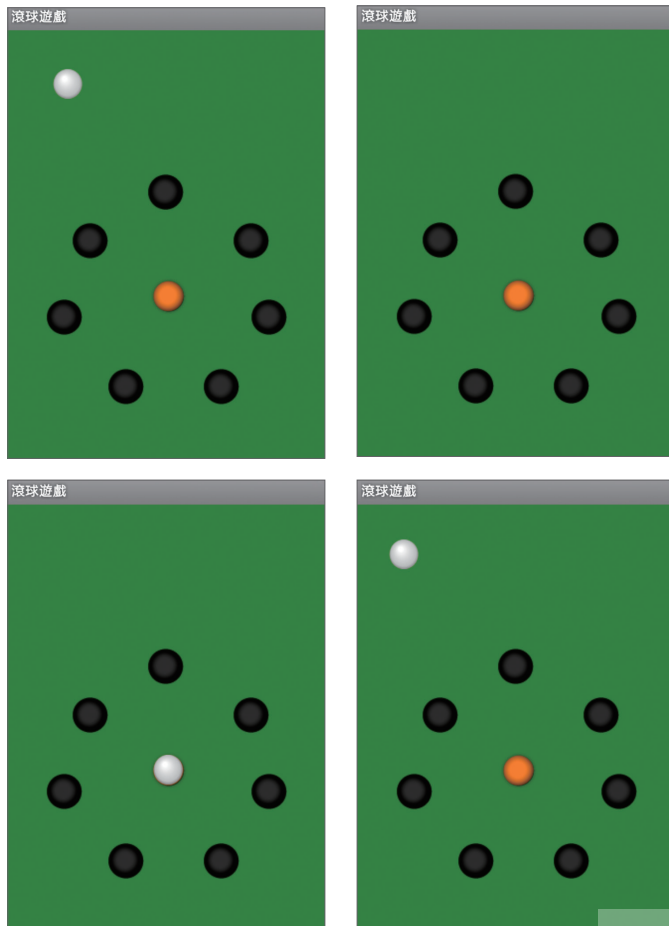
- 1 URL 儲存建立的地標。
- 2 地標顏色為藍色。
- 3 建立的地標文字。
- 4 地標的緯度、經度。
- 5 建立的地標文字，由 1 開始累加。
- 6 以指定的位置為中心顯示 Google Maps 地圖，並建立地標。

12.5 綜合練習

許多遊戲中，都會加入加速度感測器，增加遊戲的張力和精彩度，善用加速度感測器可以製作抽籤機、計步器，如果能結合 **Drawing and Animation** 元件，更能讓遊戲加分。

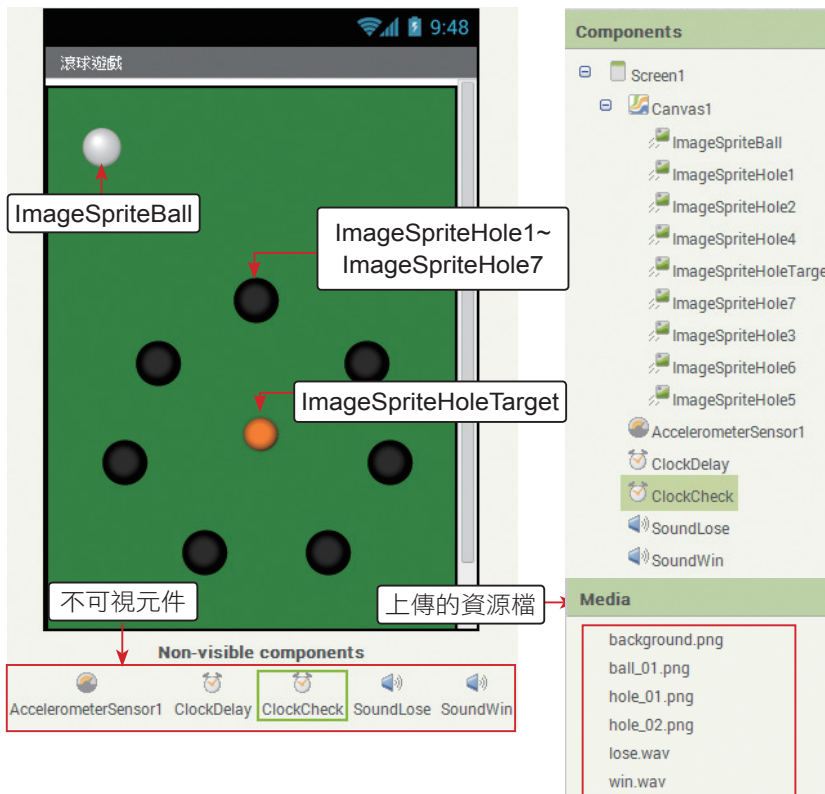
▣ 範例：滾球遊戲

將桌面上的白球以感測器控制，滾動進入中間的橘色球洞中，表示完成遊戲，並發出成功音效；若白球滾進碰到旁邊黑色球洞，表示遊戲失敗，同時發出失敗音效。當白球碰到黑色球洞會消失，經過 1 秒後，白球又會出現在左上方並繼續遊戲。(<ch12\ex_RollBall.aia>)





» 介面配置



本例中使用兩個計時器 **ClockDelay** 和 **ClockCheck**，**ClockDelay** 用以設定白球進黑色球洞後隱藏的時間，預設是 1 秒鐘，也就是說白球進入黑色球洞後會隱藏 1 秒鐘，然後再出現在螢幕上方，準備繼續遊戲。

ClockCheck 計時器則是設定多久時間檢查白球是否進洞（包含黑色球洞、橘色球洞），預設是 0.1 秒檢查一次。**ClockCheck** 計時器的 **TimerInterval** 屬性必須由經驗評估，若 **TimerInterval** 設太大，感測較不靈敏，**TimerInterval** 設太小，則又會浪費太多效能檢查上。

SoundLose、**SoundWin** 則分別播放進入黑色球洞、橘色球洞的音效。