



深入解析

圖片選擇器 元件的外觀與按鈕元件相同。

使用者選取相片後會觸發 **選擇完成** 事件，並且傳回使用者選取的相片路徑，該路徑儲存於 **選中項** 屬性中，設計者可根據取得的相片路徑在本事件做後續處理。

要啟動 **圖片選擇器** 元件的方式有兩種：第一種方式是使用者直接點選 **圖片選擇器** 元件，系統就會開啟行動裝置的相簿讓使用者選取；第二種方式是在程式拼塊中以 **打開選框** 方法啟動 **圖片選擇器** 元件。**打開選框** 方法的作用相當於使用者點選 **圖片選擇器** 元件，會開啟行動裝置的相簿讓使用者選取。

7.1.3 整合範例：照相及選取相片

▣ 範例：照相機

程式執行後，螢幕上方顯示預設圖片。按 **使用相機照相** 鈕後會開啟行動裝置的照相機，使用者照相後會將相片顯示於上方，下方則以紅色字體列出相片的儲存路徑。



按 **直接啟動選檔** 鈕會開啟行動裝置相片檔案資料夾，顯示所有相片讓使用者選取，使用者點選相片後，該相片會顯示於上方，下方則顯示相片的儲存路徑。按 **程式啟動選檔** 鈕的效果與按 **直接啟動選檔** 鈕完全相同。(<ch07\ex_Camera.aia>)



建議使用實機執行

本範例的照相功能，必須在行動裝置上執行。

» 介面配置

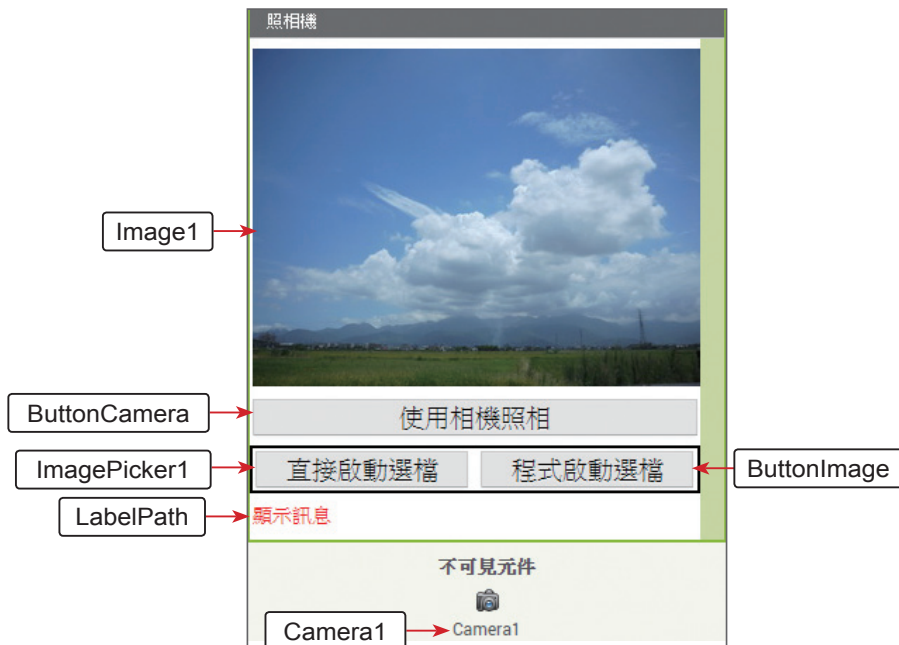
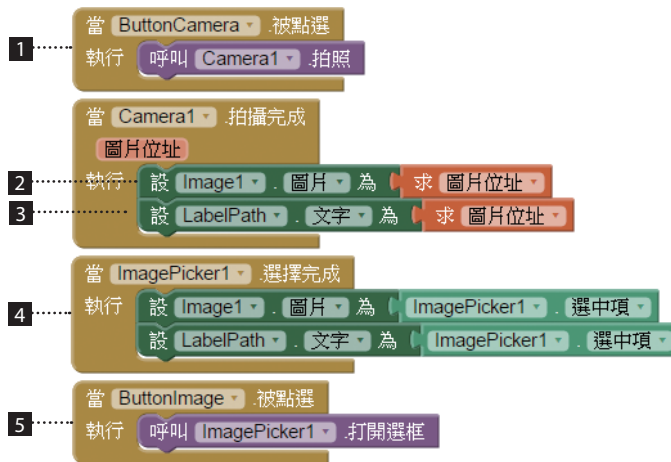




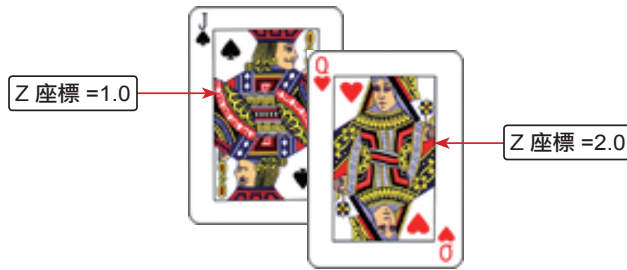
Image1 是 **圖片** 元件，用來顯示相片。為了讓頁面較為美觀，本範例預先上傳一張圖片 <img01.jpg> 做為 Image1 元件的預設圖片，即設定 Image1 元件的 **圖片** 屬性值為「img01.jpg」。

圖片選擇器 元件的外觀與按鈕完全相同，本範例中 **直接啟動選檔** 是 **圖片選擇器** 元件，用於直接開啟 **圖片選擇器** 元件；**程式啟動選檔** 是 **按鈕** 元件，用於以程式拼塊開啟 **圖片選擇器** 元件。

» 程式拼塊



- 1 使用者按 **使用相機照相** 鈕後會開啟行動裝置的照相機。
- 2 使用者照相後將相片顯示於 Image1 元件中。
- 3 使用者照相後顯示相片存放路徑。
- 4 使用者按 **直接啟動選檔** 鈕並選取相片後觸發本事件：將相片顯示於 Image1 元件中並顯示相片存放路徑。
- 5 使用者按 **程式啟動選檔** 鈕後以程式開啟相簿讓使用者選取。



在介面配置階段，建立的 **圖片精靈** 或 **球形精靈** 元件，預設的 **Z 座標** 屬性為 1.0，如果建立多個 **圖片精靈** 或 **球形精靈** 元件，所有元件的 **Z 座標** 屬性都是 1.0。由於所有的 **圖片精靈** 或 **球形精靈** 元件的 **Z 座標** 屬性都是 1.0，App Inventor 2 會將較後建立或較後拖曳的 **圖片精靈** 或 **球形精靈** 元件放置在最上層。

也可以在程式執行階段，設定 **Z 座標** 屬性將該元件移至上層。

例如：設定 **圖片精靈 1.Z 座標 = 2.0** 則 **圖片精靈 1** 將移至其他 **圖片精靈** 屬性 **Z 座標 = 1.0** 元件的上層。

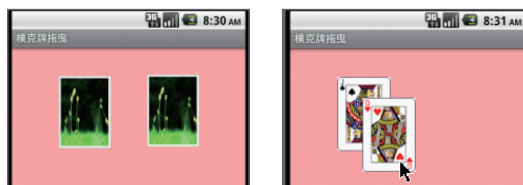
當所有 **圖片精靈** 或 **球形精靈** 元件的 **Z 座標** 屬性都相同時，只要在程式執行階段再以原來的 **Z 座標** 值重新設定一次，該元件將會移至最上層。

例如：在介面配置階段設定 **圖片精靈 1.Z 座標 = 1.0**、**圖片精靈 2.Z 座標 = 1.0**、**圖片精靈 3.Z 座標 = 1.0**。

在執行階段，設定 **圖片精靈 1.Z 座標 = 1.0**，則 **圖片精靈 1** 元件會移至最上層，設定 **圖片精靈 2.Z 座標 = 1.0**，則 **圖片精靈 2** 元件會移至最上層。同理，設定 **圖片精靈 3.Z 座標 = 1.0**，則 **圖片精靈 3** 元件將會移至最上層。

▣ 範例：撲克牌掀牌和拖曳

兩個 **圖片精靈** 元件分別表示為撲克牌 J 和 Q，程式剛開始都顯示牌背，按下撲克牌即可將牌掀開，也可拖曳撲克牌，被拖曳的牌會移至最上層顯示。
(<ch09\ex_DragPoker.aia>)



» 介面配置

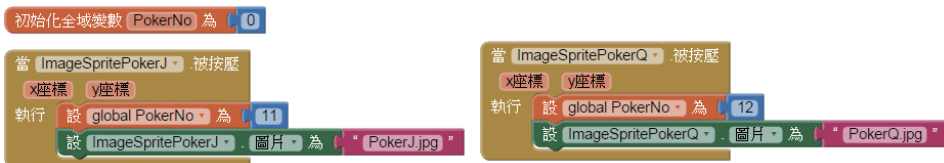


» 使用元件及其重要屬性

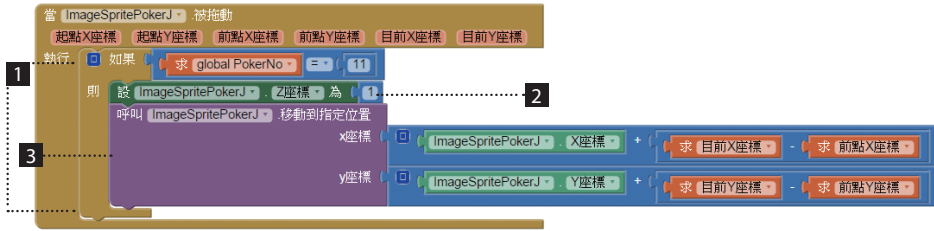
名稱	屬性	說明
Canvas1	背景顏色：粉紅色 寬度：填滿 高度：480 像素	設定畫布背景色和大小。
ImageSpritePokerJ	方向：0 間隔：100 圖片：PokerBackground.jpg 旋轉：true 速度：0 高度：自動 寬度：自動	設定 ImageSpritePokerJ 的背景圖為 <PokerBackground.jpg>。
ImageSpritePokerQ	同上	同上。

» 程式拼塊

1. 按下撲克牌將牌掀開，並以 `PokerNo` 記錄撲克牌編號，撲克牌 J、Q 編號分別為 11 和 12。



2. 判斷拖曳牌是否為撲克牌 J，成立時才拖曳該牌。當兩張牌重疊時，如果不檢查將會產生誤判。加上 `PokerNo` 的判斷即可明確拖曳指定的牌。



- 1 如果編號 11，拖曳的牌為撲克牌 J。
- 2 將拖曳的牌移到最上層。
- 3 拖曳撲克牌。

ImageSpritePokerQ. 被拖動 事件和 **ImageSpritePokerJ. 被拖動** 事件相似，但必須檢查編號為 12，不再贅述。

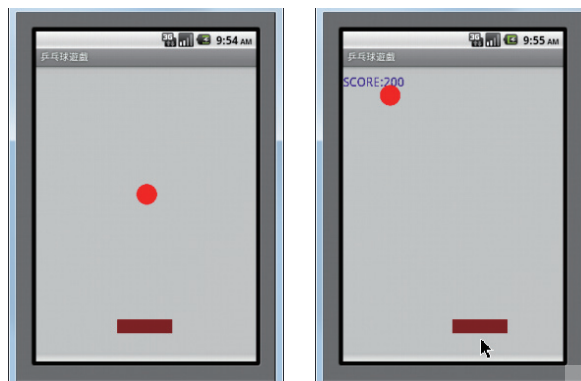
9.3 綜合練習

圖片精靈 和 **球形精靈** 元件，可說是最適合設計遊戲的元件，當然，它有個限制，必須配合 **畫布** 使用，當您熟悉這些元件的特性後，是否有個衝勁，想要馬上現學現賣，立即完成一個小小的遊戲呢？

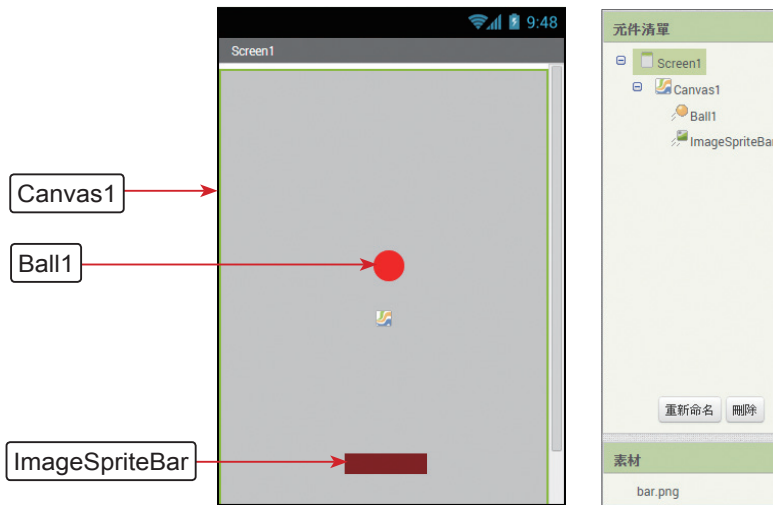
接下來，我們以一個較為完整的範例，來展現 **圖片精靈** 和 **球形精靈** 元件，同時，程式中也加入一些小技巧，增加遊戲張力。

▣ 範例：乒乓球遊戲

用手指按下紅色的球開始發球，遊戲者可左、右拖曳紅色擋板，當球碰到紅色擋板可得 50 分並反彈，碰到 **畫布** 的右、上、左邊緣時也會反彈但不會得分，碰到 **畫布** 下邊緣，則結束遊戲並將紅色球移到螢幕中央。(〈ch09\ex_PingPong.aia〉)



» 介面配置



» 使用元件及其重要屬性

名稱	屬性	說明
Canvas1	寬度：填滿 高度：480 像素	設定畫布大小寬度填滿整個螢幕、高度 480 pixels。
Ball1	方向：0 間隔：100 畫筆顏色：紅色 速度：0 半徑：15	設定 Ball1 的背景色為紅色。
ImageSpriteBar	圖片：bar.png 寬度：80 像素	擋板背景圖為 <bar.png>。

» 程式拼塊

1. 建立 Score 記錄遊戲得分。

初始化全域變數 Score 為 0

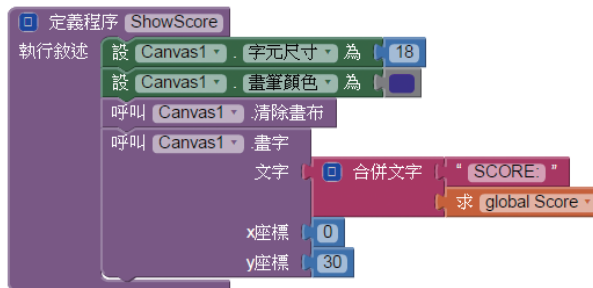


2. 按下 **紅色球** 開始將球發出。

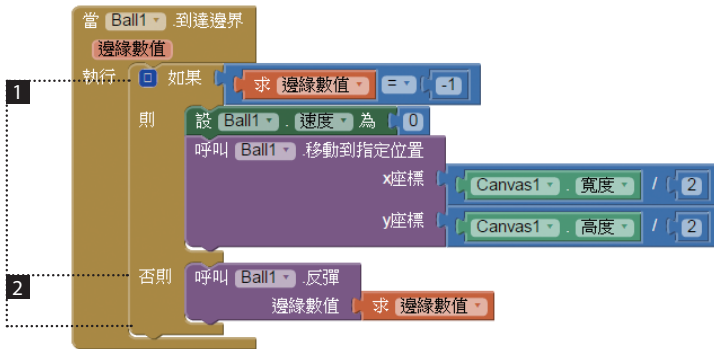


- 1 按下 **紅色球** 開始發球。
- 2 程式開始，設定得分為 0。
- 3 以自訂程序 ShowScore 顯示遊戲得分。
- 4 將 **紅色球** 移至中央。
- 5 發球的方向為 30~120 中以亂數取得的角，球速為 30。

3. 自訂程序 ShowScore 以藍色、字型大小為 18，顯示遊戲得分。

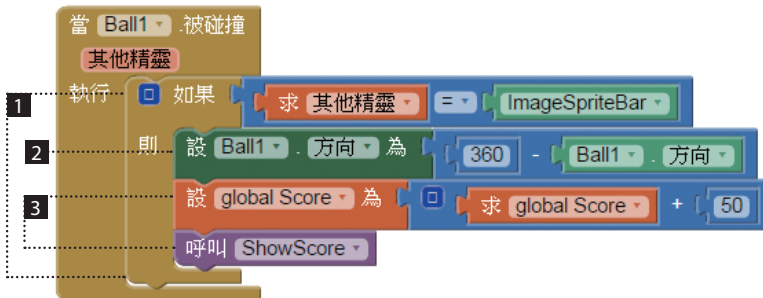


4. 當球碰到邊界的處理。

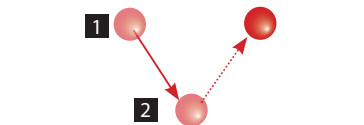


- 1 如果是碰到下邊界，表示遊戲結束，將球停止移動，並移到螢幕中央準備下次再進行遊戲。
- 2 如果是碰到右、上、左邊界，將球反彈。

5. 當球碰到擋板，得分加 50 分。



- 1 判斷球是否碰到擋板。
- 2 將球反彈。球移動後碰撞到擋板再反彈的方向計算公式為： $\text{反彈方向} = (360 - \text{移動方向})$ 。例如：球朝著 300° 方向移動，在碰撞到擋板後反彈方向為 $(360 - 300 = 60) 60^\circ$ 。
- 3 得分加 50 分，更新得分。

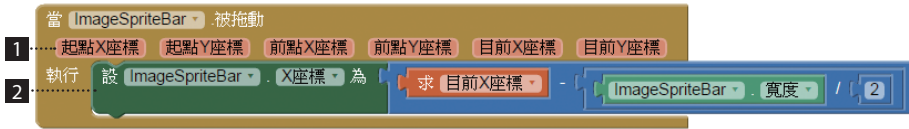


球先在 1 位置朝右下角 300° 方向移動，在 2 碰到擋板後就往右上角 60° 方向移動。

- 3 得分加 50 分，更新得分。



6. 拖曳擋板的處理。



- 1 接收的參數，其中 (起點 X 座標, 起點 Y 座標) 為拖曳的起點，在拖曳過程 (前點 X 座標, 前點 Y 座標) 和 (目前 X 座標, 目前 Y 座標) 會依拖曳位置而改變，(前點 X 座標, 前點 Y 座標) 為上一次的觸碰點，(目前 X 座標, 目前 Y 座標) 為目前的觸碰點。
- 2 將 擋板 的中心點移到目前的觸碰點 目前 X 座標 位置上，也就是說 擋板 的中心點會隨著觸碰點移動。