



5.3 綜合演練：餵蝴蝶遊戲

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？

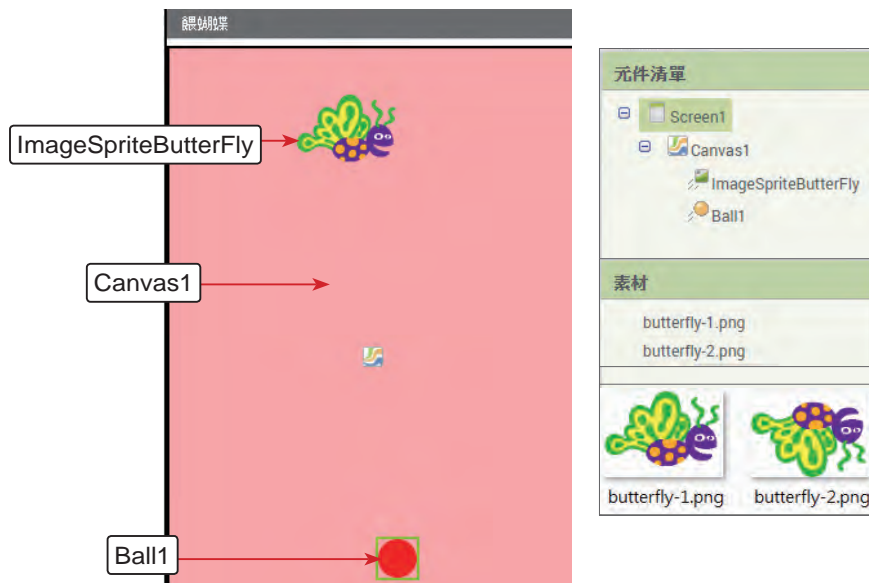
接下來，我們以一個較為完整的範例，來展現 **圖片精靈** 和 **球形精靈** 元件，同時，程式中也加入一些小技巧，增加遊戲張力。

▽ 範例：餵蝴蝶遊戲

饑餓的蝴蝶在空中左、右不斷飛翔，蝴蝶碰到 **畫布** 的左、右邊緣時會返回飛翔。按下方紅色的食物圖示，食物會往上前進直到碰到上邊界才消失，蝴蝶吃到食物可得 1 分，並將食物再放置於下方，準備再進行遊戲。(<ch05\ex_FeedButterFly.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 背景顏色：粉紅色 高度：480 像素	設定畫布大小寬度填滿整個螢幕、高度 480 像素。
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。
ImageSpriteButterFly	圖片：butterfly-1.png 方向：0 間隔：100 速度：0 寬度：80 像素 高度：60 像素	碰到左、右邊界會分別更改背景圖為 <butterfly-1.png>、<butterfly-2.png>。



» 程式拼塊

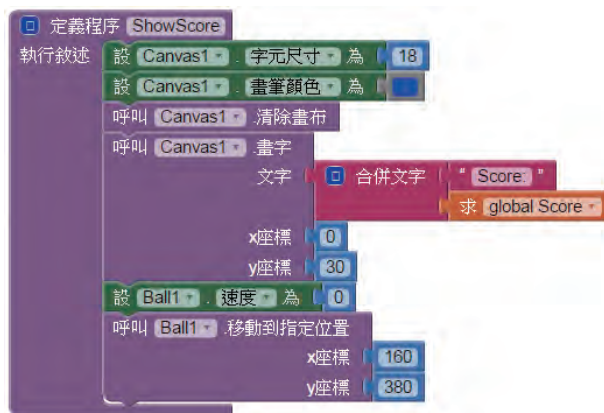
1. 建立 Score 記錄遊戲得分。



2. 程式初始化。



- 1 將 紅色食物 暫停。
 - 2 設定蝴蝶飛翔的速度為 20。
 - 3 以自訂程序 ShowScore 顯示遊戲得分。
3. 自訂程序 ShowScore 以藍色、字型大小為 18，顯示遊戲得分，同時將 紅色食物 暫停並移回初始的位置，準備再進行遊戲。



4. 按下 紅色食物 開始將 紅色食物 往上發出。



- 1 設定 紅色食物 的速度為 50。
- 2 設定 紅色食物 的方向為向上。

5. 當 **紅色食物** 碰到邊界，將 **紅色食物** 暫停，並移到螢幕下方準備再進行遊戲。



6. 當蝴蝶碰到左、右邊界的處理。



- 1 蝴蝶碰到左邊界，設定背景圖片為向右飛翔的圖檔。
- 2 蝴蝶碰到右邊界，設定背景圖片為向左飛翔的圖檔。
- 3 將蝴蝶反彈，讓蝴蝶反向飛翔。

7. 蝴蝶碰到 **紅色食物** 的處理。



- 1 如果蝴蝶碰到元件會觸發此事件。
- 2 如果碰到元件是 **紅色食物**，得分加 1 分並更新分數。

6.3 打雪怪遊戲設計

在這一章中，我們選擇一個難度不高，但介面卻非常吸引人的專題「打雪怪遊戲」當做讀者初次嘗試專題的體驗。同時為了讓讀者學習更順利，先花了一些篇幅探討本專題使用的元件。

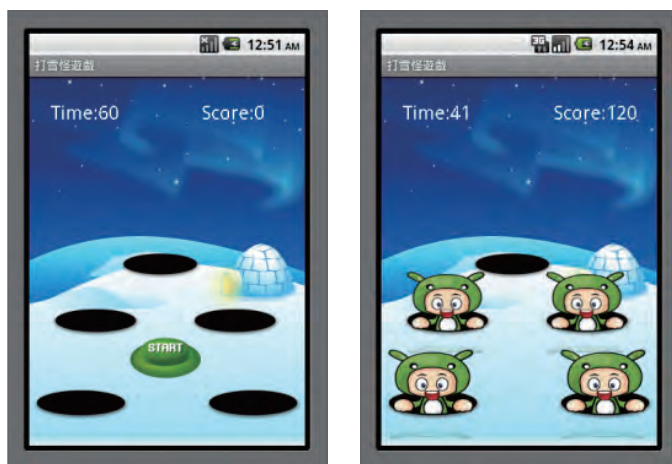
6.3.1 專題發想

「打雪怪遊戲」仍然保持打地鼠遊戲的架構，但整個遊戲場景是在冰天雪地中，月夜裡雪人會不斷出沒。遊戲者必須以最敏捷的動作，正確擊中從地洞中突然冒出的雪人，否則雪人隨時又會立即鑽入地洞中。

6.3.2 專題總覽

按下 **Start** 按鈕開始計時 60 秒，得分從 0 開始。遊戲進行中雪人會不斷出沒，如果您擊中雪人得分會加 10 分。

專題路徑：<ch06\mypro_MouseGame.aia>



建議使用實機執行

這個專題在實機執行時，觸控的效果極佳，然而在模擬器上執行時觸控的效果卻不如預期，建議您在實機執行。



6.3.3 介面配置



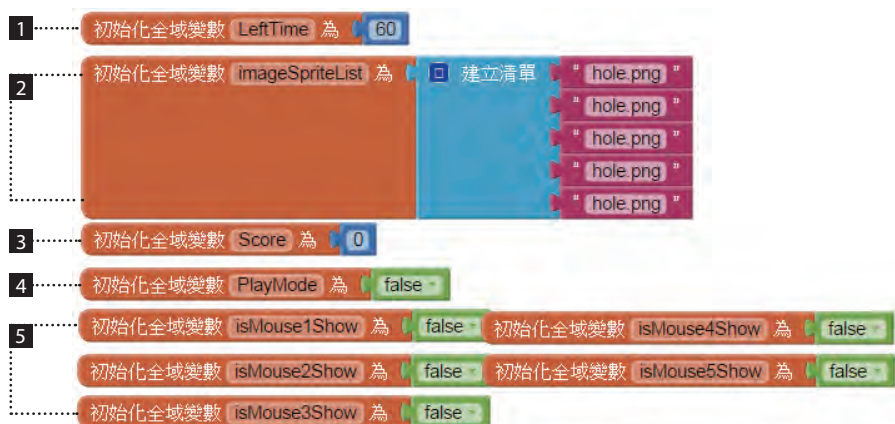
使用元件及其重要屬性

元件類別	名稱	屬性	說明
Screen	Screen1	標題：打雪怪遊戲 圖示：icon_MouseGame.png 畫面方向：鎖定直式畫面	設定應用程式標題、圖示，螢幕方向為直向。
畫布	Canvas1	背景圖片：background.png	遊戲的場景。
圖片精靈	ImageSpritePlay	圖片：start.png	遊戲開始按鈕。
圖片精靈	ImageSpriteHole1~ ImageSpriteHole5	圖片：hole.png	顯示地洞。
音訊播放器	PlayerBackground	來源文件：music.mp3	播放背景音效。
音效	SoundGain	來源文件：hit.wav 最小間隔：100	播放得分音效。
計時器	ClockTimer	計時間隔：1000	遊戲計時。
計時器	ClockMouse1~ ClockMouse5	無	以動態方式設定每一個地洞中雪人顯示的時間。

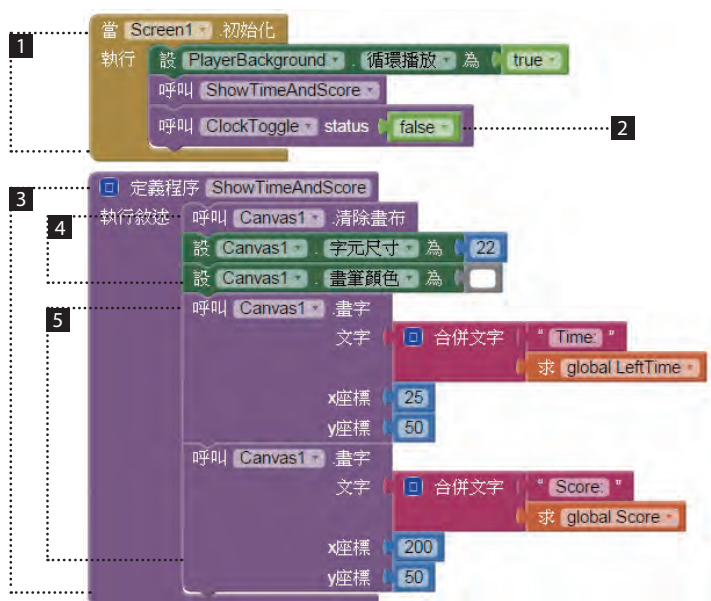
6.3.4 專題分析和程式拼塊說明

1. 建立全域變數 LeftTime、imageSpriteList、Score、PlayMode 和 isMouse1Show ~ isMouse5Show。

LeftTime=60 記錄遊戲時間，imageSpriteList 清單記錄目前地洞躲藏的角色，Score=0 記錄得分，playMode=false 代表遊戲尚未開始，isMouse1Show ~ isMouse5Show 分別記錄各地洞是否已出現雪人。



- 1 LeftTime=60 記錄遊戲共有 60 秒鐘。
 - 2 imageSpriteList 為一個 文字 型別的陣列，共有 5 個元素，是用來記錄目前地洞躲藏的角色，初始值都是 <hole.png> 表示目前的圖示是地洞。遊戲執行後 imageSpriteList 的狀態有可能是 雪人：goodguy.png 或 地洞：hole.png。
 - 3 Score=0 記錄得分從 0 開始。
 - 4 PlayMode=false 表示遊戲尚未開始，當按下 **Start** 按鈕，則會將 PlayMode 設為 true，表示遊戲已在進行中。
 - 5 isMouse1Show ~ isMouse5Show 分別記錄各地洞是否已出現雪人，預設為 false 代表尚未出現。
2. 設定背景音樂循環播放，遊戲尚未開始，背景音樂暫停播放，以自訂程序 ClockToggle 停止所有的計時器，同時也以自訂程序 ShowTimeAndScore 顯示遊戲剩餘的時間和得分。



- 1** 設定背景音樂循環播放，程式初始背景音樂暫停播放，停止所有的計時器。
 - 2** 以自訂程序 ClockToggle 停止所有的計時器。
 - 3** 自訂的程序 ShowTimeAndScore 顯示遊戲剩餘的時間和得分。
 - 4** 清除 畫布 並設定畫筆的大小為 22 像素、顏色為白色。
 - 5** 顯示遊戲剩餘的時間和得分，程式初始 **Time : 60**、**Score : 0**。
3. 按下 **Start** 按鈕，計時 60 秒，得分從 0 開始，啟動背景音樂並將 **Start** 按鈕隱藏起來，同時設定 **PlayMode=true** 記錄遊戲已經開始。記錄遊戲時間的計時器和所有啟動的雪人出現的計時器全部啟動。



- 1** 計時 60 秒，得分從 0 開始。

- 2 以自訂的程序 ShowTimeAndScore 顯示遊戲剩餘的時間和得分。
 - 3 將 **Start** 按鈕隱藏起來。
 - 4 啟動背景音樂，同時設定 PlayMode=true 記錄遊戲已經開始。
 - 5 以自訂的程序 ClockToggle 將記錄遊戲時間的計時器、所有啟動雪人出現的計時器全部啟動。
4. 自訂的程序 ClockToggle 中接收 status 參數，並依 status 參數設定計時器，當 status 為 true 時將所有計時器全部啟動，而當 status 為 false 時將所有計時器全部停止。



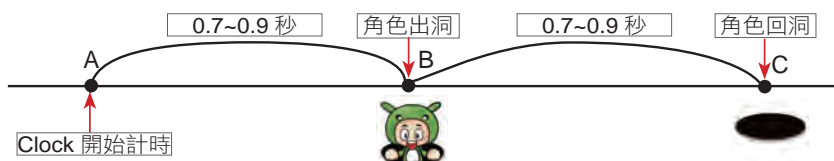
- 1 自訂的程序 ClockToggle 中接收參數 status。
 - 2 當 status 為 true 時將所有計時器全部啟動，而當 status 為 false 時將所有計時器全部停止。ClockTimer 計時器記錄遊戲時間，ClockMouse1~ClockMouse5 則啟動雪人開始出現，ClockMouse1 控制 ImageSpriteHole1、ClockMouse2 控制 ImageSpriteHole2，依此類推。
5. ClockTimer 的計時事件，每一秒鐘執行一次，當遊戲終了停止背景音樂，並以 PlayMode=false 記錄遊戲已停止，同時再將 **start** 按鈕顯示，所有的計時器也全部停止計時。





- 1 將遊戲時間減 1 並更新顯示。
 - 2 如果遊戲時間遞減至 0 則停止背景音樂，並設定 `PlayMode=false` 記錄遊戲已停止，同時再將 **start** 按鈕顯示，並以 `ClockToggle(false)` 將所有的計時器全部停止計時。
6. **ClockMouse1. 計時 ~ ClockMouse5. 計時** 共 5 個事件是這個專題最核心的部分，因為 5 個事件的原理都相同，我們只列出 **ClockMouse1. 計時** 的程式拼塊。**ClockMouse1. 計時** 事件會判斷 `isMouse1Show` 的狀態，如果是 `true`，表示前次的狀態是雪人，因此這次要顯示 `ImageSpriteHole1` 的 **圖片** 屬性為地洞 (`hole.png`); 如果是 `false`，表示前次的狀態是地洞，因此這次要顯示雪人 (`goodguy.png`)。

對每個角色而言，都使用個別的 **計時器** 來計時，第一個計時器名稱為 `ClockMouse1`，它會在 A 點開始計時，至 B 點角色 (雪人) 會冒出地洞，到了 C 點，角色就躲回地洞了，這樣就是一個循環。



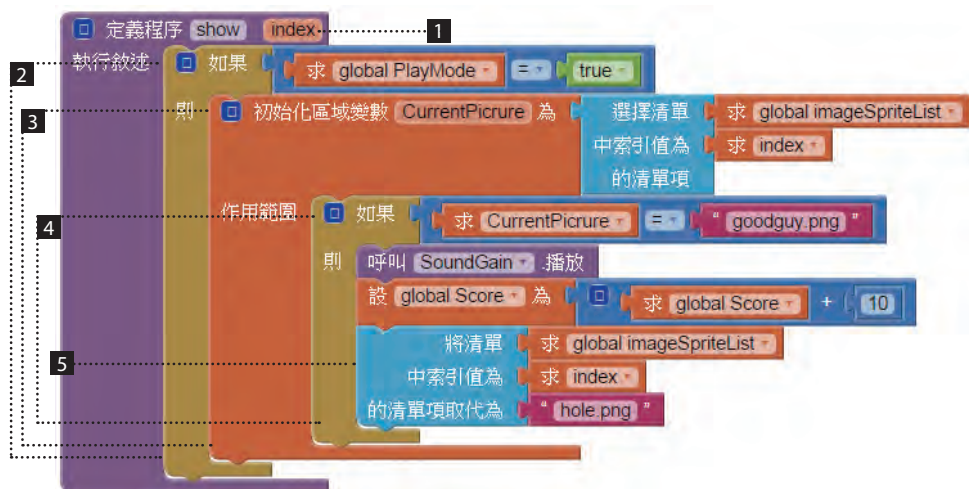
還有一點非常重要，將目前地洞的背景圖，依序存在 `imageSpriteList` 清單中，即 `ImageSpriteHole1` 的 **圖片** 屬性存在 `imageSpriteList(1)` 中，所以實際上 `ImageSpriteHole1` 的 **圖片** 屬性有可能是 **雪人**：`goodguy.png` 或 **地洞**：`hole.png` 兩種狀態。這樣只要判斷 `imageSpriteList` 清單就知道每一個 `ImageSpriteHole1~ImageSpriteHole5` 的狀態。



- 1 每次都重設 ClockMouse1 下次執行的時間為 0.7~0.9 秒，遊戲者可以自行調整這個時間控制遊戲的節奏。
 - 2 如果 isMouse1Show = false，就顯示雪人 (goodguy.png)，並設定 isMouse1Show = true，讓雪人和地洞輪流地出現。
 - 3 如果 isMouse1Show = true，就顯示 ImageSpriteHole1 的圖片屬性為地洞 (hole.png)，並設定 isMouse1Show = false。
 - 4 將 ImageSpriteHole1 的圖片屬性存至對應的 imageSpriteList 清單的第一個元素中。
7. 當碰觸雪人 (ImageSpriteHole1~ImageSpriteHole5) 會觸發 被按壓 事件，因為 5 個 被按壓 事件原理都相同，我們只列出 **ImageSpriteHole1. 被按壓** 事件。這些事件會執行自訂的 show() 程序，並分別傳參數 1~5，即 show(1)~show(5)。



8. 自訂的 show()，主要依據 index 索引值，取得 imageSpriteList 清單元素的字串內容至 CurrentPicture 區域變數，根據這個字串即可分辨按下角色是否是雪人。



- 1 接收參數，index 參數值由 1~5。
- 2 如果 PlayMode=true，表示遊戲已在進行中，觸控才作處理。