

# 05



## 繪圖與動畫

畫布屬於繪圖動畫類別元件，畫布相當於一個空白畫布，可以在畫布上繪製點、直線、圓、文字等圖形，也可以將畫布的圖形存檔。

圖像精靈和球形精靈屬於繪圖動畫類別元件，它是 **AppInventor2** 為動畫和遊戲所量身打造的元件，使用時必須配合畫布元件。



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

當所有 圖像精靈 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 圖像精靈 1.Z 座標 =1.0 、圖像精靈 2.Z 座標 =1.0 、圖像精靈 3.Z 座標 =1.0 。

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

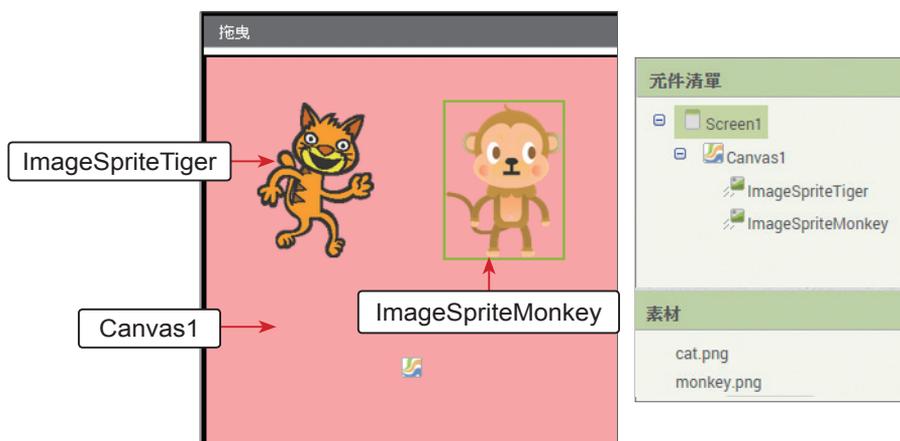
### ▣ 範例：物件拖曳

兩個 圖像精靈 元件分別為老虎和猴子，按下老虎和猴子可以將之拖曳，被拖曳的物件會移至最上層顯示。

(<ch05\ex\_DragMe.aia>)



### » 介面配置





## » 程式拼塊

1. 按下老虎和猴子，分別以 No 記錄編號，老虎和猴子編號分別為 1、2。

```
初始化全域變數 No 為 0  
當 ImageSpriteTiger 被壓下  
  x座標 y座標  
  執行 設置 global No 為 1  
當 ImageSpriteMonkey 被壓下  
  x座標 y座標  
  執行 設置 global No 為 2
```

2. 判斷拖曳者是否為老虎，成立時才拖曳該物件。當老虎和猴子重疊時，如果不檢查將會產生誤判，加上 No 的判斷即可明確拖曳指定的物件。

```
當 ImageSpriteTiger 被拖曳  
  起點X座標 起點Y座標 前點X座標 前點Y座標 當前X座標 當前Y座標  
  執行 1 如果 取 global No 為 1  
    則 2 設 ImageSpriteTiger 的 z座標 為 1  
        呼叫 ImageSpriteTiger 移動到指定位置  
          x座標 取 ImageSpriteTiger 的 X座標 + 取 當前X座標 - 取 前點X座標  
          y座標 取 ImageSpriteTiger 的 Y座標 + 取 當前Y座標 - 取 前點Y座標  
  3
```

- 1 如果編號 1，拖曳的物件是老虎。
  - 2 將拖曳的物件到最上層。
  - 3 拖曳物件。
3. **ImageSpriteMonkey. 被拖曳** 事件和 **ImageSpriteTiger. 被拖曳** 相似，但必須檢查編號為 2，不再贅述。

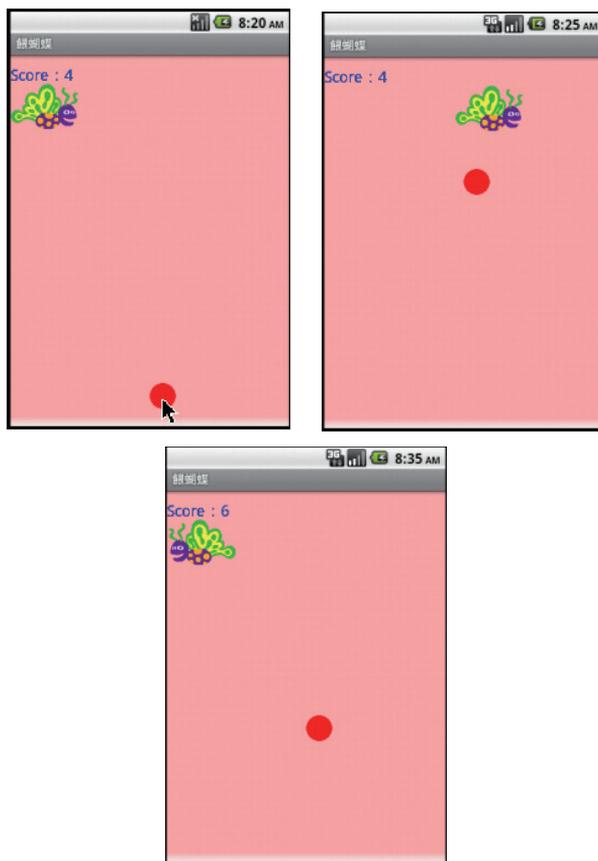
## 5.3 綜合演練：餵蝴蝶遊戲

圖像精靈 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，是否有股衝勁，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 圖像精靈 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。

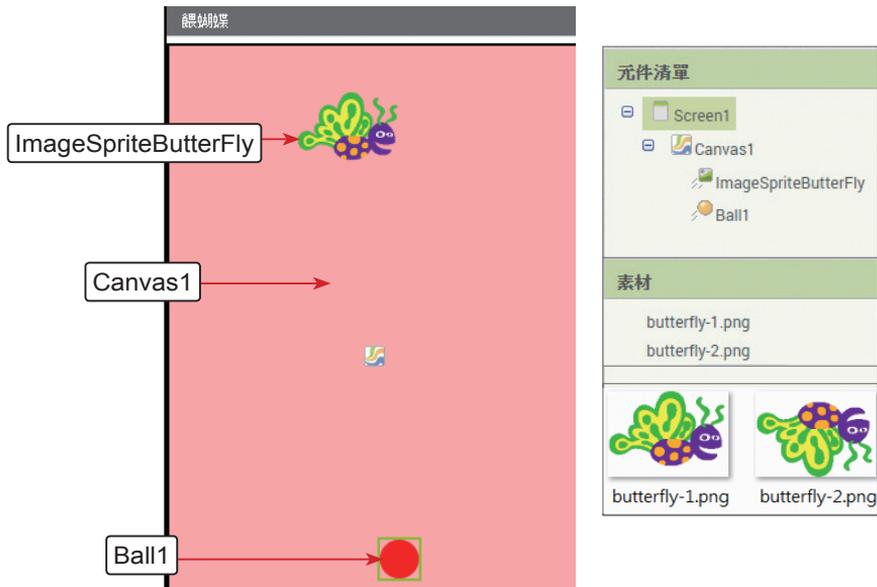
### ▣ 範例：餵蝴蝶遊戲

飢餓的蝴蝶在空中左、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<br>背景顏色：粉紅色<br>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                | 指向：0<br>間隔：100<br>畫筆顏色：紅色<br>速度：0<br>半徑：15                           | 設定 Ball1 的畫筆顏色為紅色。                                    |
| ImageSpriteButterFly | 圖片：butterfly-1.png<br>指向：0<br>間隔：100<br>速度：0<br>寬度：80 像素<br>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。

初始化全域變數 Score 為 0

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、字體大小為 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。

```
當 Ball1 到達邊界
  邊緣數值
  執行
    設 Ball1 速度 為 0
    呼叫 Ball1 移動到指定位置
      x座標 160
      y座標 380
```

6. 當蝴蝶碰到左、右邊界的處理。

```
當 ImageSpriteButterFly 到達邊界
  邊緣數值
  執行
    1 如果 取 邊緣數值 = -3
      則 設 ImageSpriteButterFly 圖片 為 "butterfly-1.png"
    2 否則，如果 取 邊緣數值 = 3
      則 設 ImageSpriteButterFly 圖片 為 "butterfly-2.png"
    3 呼叫 ImageSpriteButterFly 反彈
      邊緣數值 取 邊緣數值
```

- 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。

```
當 ImageSpriteButterFly 碰撞
  其他精靈
  執行
    1 如果 取 其他精靈 = Ball1
      2 則 設置 global Score 為 取 global Score + 1
      呼叫 ShowScore
```

- 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。

## 15.2 打雪怪遊戲設計

在這一章中，我們選擇一個較完整的專案「打雪怪遊戲」，除了加入計分、音效，介面也非常吸引人，同時也以亂數控制遊戲的精彩度。

### 15.2.1 專案發想

「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。

### 15.2.2 專案總覽

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。

專案路徑：`<ch15\mypro_MouseGame.aia>`



### 建議使用實機執行

這個專案在實機執行時，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。



## 15.2.3 介面配置

The screenshot shows the App Inventor 2 interface for a game titled "打雪怪遊戲". The main canvas displays a snowy landscape with an igloo, a "START" button, and several holes. Callouts identify the following elements:

- ImageSpriteHole1 ~ ImageSpriteHole5: Pointing to the holes in the snow.
- ImageSpritePlay: Pointing to the "START" button.

The "元件清單" (Component List) on the right shows the following structure:

- Screen1
  - Canvas1
    - ImageSpriteHole4
    - ImageSpriteHole5
    - ImageSpriteHole1
    - ImageSpriteHole2
    - ImageSpriteHole3
    - ImageSpritePlay
  - PlayerBackground
  - SoundGain
  - ClockMouse1
  - ClockMouse2
  - ClockMouse3
  - ClockMouse4
  - ClockMouse5
  - ClockTimer

The "非可見元件" (Non-visible elements) palette at the bottom shows icons for PlayerBackground, SoundGain, and five ClockMouse components.

## 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<br>圖示：icon_MouseGame.png<br>螢畫方向：鎖定直式畫面<br>狀態欄顯示：取消核選 | 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。                |
| 圖像精靈   | ImageSpritePlay                       | 圖片：start.png   | 遊戲開始按鈕。               |
| 圖像精靈   | ImageSpriteHole1~<br>ImageSpriteHole5 | 圖片：hole.png  | 顯示地洞。                 |
| 音樂播放器  | PlayerBackground                      | 來源：music.mp3   | 播放背景音效。               |
| 音效     | SoundGain                             | 來源：hit.wav<br>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~<br>ClockMouse5           | 無  | 以動態方式設定每一個地洞中雪人顯示的時間。 |

## 15.2.4 專案分析和程式拼塊說明

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



- 1 LeftTime=60 記錄遊戲共有 60 秒鐘。
  - 2 imageSpriteList 為一個文字型別的清單，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 = 假 表示遊戲尚未開始，當按下 **Start** 按鈕，則會將 PlayMode 設為 真，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 假 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。



The screenshot shows the following code blocks:

- 1** 當 Screen1 初始化 執行:
  - 設 PlayerBackground 循環播放 為 真
  - 呼叫 ShowTimeAndScore
  - 呼叫 ClockToggle status 假
- 3** 定義程序 ShowTimeAndScore:
  - 4** 執行 呼叫 Canvas1 清除畫布
  - 設 Canvas1 字體大小 為 22
  - 設 Canvas1 畫筆顏色 為 白
  - 5** 呼叫 Canvas1 繪製文字:
    - 文字 合併文字 "Time:" 取 global LeftTime
    - x座標 25
    - y座標 50
  - 呼叫 Canvas1 繪製文字:
    - 文字 合併文字 "Score:" 取 global Score
    - x座標 200
    - y座標 50

- 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 = 真** 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。

The screenshot shows the following code blocks:

- 當 ImageSpritePlay 被壓下 執行:
  - 1** 設置 global LeftTime 為 60
  - 設置 global Score 為 0
  - 2** 呼叫 ShowTimeAndScore
  - 3** 設 ImageSpritePlay 可見性 為 假
  - 4** 呼叫 PlayerBackground 開始
  - 設置 global PlayMode 為 真
  - 5** 呼叫 ClockToggle status 真

- 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 = 真 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 真 時將所有計時器全部啟動，而當 status 為 假 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 真 時將所有計時器全部啟動，而當 status 為 假 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 = 假 記錄遊戲已停止，同時再將 **start** 按鈕顯示，所有計時器也全部停止計時。



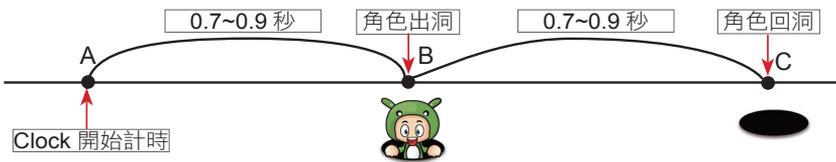


## App Inventor 2 初學特訓班

- 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 = 假 記錄遊戲已停止，同時再將 **start** 按鈕顯示，並以 ClockToggle( 假 ) 將所有計時器全部停止計時。

6. **ClockMouse1. 計時 ~ ClockMouse5. 計時** 共 5 個事件是這個專案最核心的部分，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 **真**，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 **假**，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。



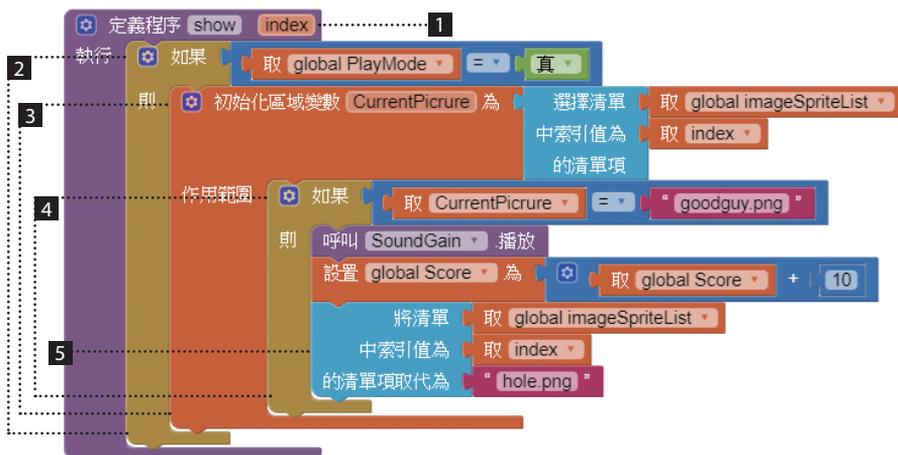
還有一點非常重要，將目前地洞的背景圖，依序存在 imageSpriteList 清單中，即 ImageSpriteHole1 的 **圖片** 屬性存在 imageSpriteList(1) 中，所以實際上 ImageSpriteHole1 的 **圖片** 屬性有可能是 **雪人 : goodguy.png** 或 **地洞 : hole.png** 兩種狀態。這樣只要判斷 imageSpriteList 清單就知道每一個 ImageSpriteHole1~ImageSpriteHole5 的狀態。



- 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 = 假，就顯示雪人 (goodguy.png)，並設定 isMouse1Show = 真，讓雪人和地洞輪流地出現。
  - 3 如果 isMouse1Show = 真，就顯示 ImageSpriteHole1 的 圖片 屬性為地洞 (hole.png)，並設定 isMouse1Show = 假。
  - 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 被壓下 事件，因為 5 個 被壓下 事件原理都相同，我們只列出 **ImageSpriteHole1. 被壓下** 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 = 真，表示遊戲已在進行中，觸控才作處理。