

# Unit 9 繪圖與動畫

作者：劉映辰

## 畫布元件 (Canvas)

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、文字等圖形。此外，AI2 經常使用畫布配合圖片精靈和球形精靈元件，設計含有動畫的遊戲程式。

## 畫布元件介紹

畫布相當於一個空白畫布，使用背景顏色可以設定畫布背景顏色，也可以使用背景圖片設定背景圖。

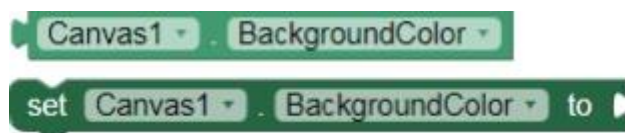
畫布上可以繪製點、直線、實心圓、文字等圖形，同時可將畫布上繪製的圖檔存檔。

畫布的座標計算是以向右為正、向下為正，並以畫布左上角為(0, 0)基準點，所計算的相對座標。

畫布只允許圖片精靈和球形精靈兩種元件。

## 畫布常用元件

BackgroundColor(背景顏色): 設定畫布的背景顏色。



DrawLine: 在畫布上畫出一條直線，X1Y1 為起始位置，X2Y2 為結束位置。



Save: 將畫布當下的狀態擷取成一張圖片，並儲存於裝置的外部 SD 記憶卡內，並回傳位置

```
call Canvas1 .Save
```

SaveAs: 和 Save 一樣是擷取目前畫布狀態為一張圖片，但此方法副檔名必須使用.JPEG、.JPG 或.PNG。

```
call Canvas1 .SaveAs  
fileName
```

DrawTextAtAngle: 在畫布的指定座標上顯示文字，並指定旋轉的角度(Angle)。

```
call Canvas1 .DrawTextAtAngle  
text 要顯示的文字  
x X座標  
y Y座標  
angle 旋轉的角度，從0開始為水平
```

SetBackgroundPixelColor: 設置指定座標處(x、y)的顏色。

```
call Canvas1 .SetBackgroundPixelColor  
x  
y  
color
```

Flung: 回傳使用者手滑過畫布後的一些數據(例如:x、y 座標、速度....等)，flungSprite 代表滑動過程中是否碰到圖片或球形精靈。

```
when Canvas1 .Flung  
x y speed heading xvel yvel flungSprite  
do
```

Dragged: 當使用者作拖拉動作時，觸控點會從 (prevX,prevY) 移動到 (currentX,currentY)(startX, startY 為起使座標)，draggedAnySprite 表示拖曳過程是否有觸碰到圖片或球形精靈。



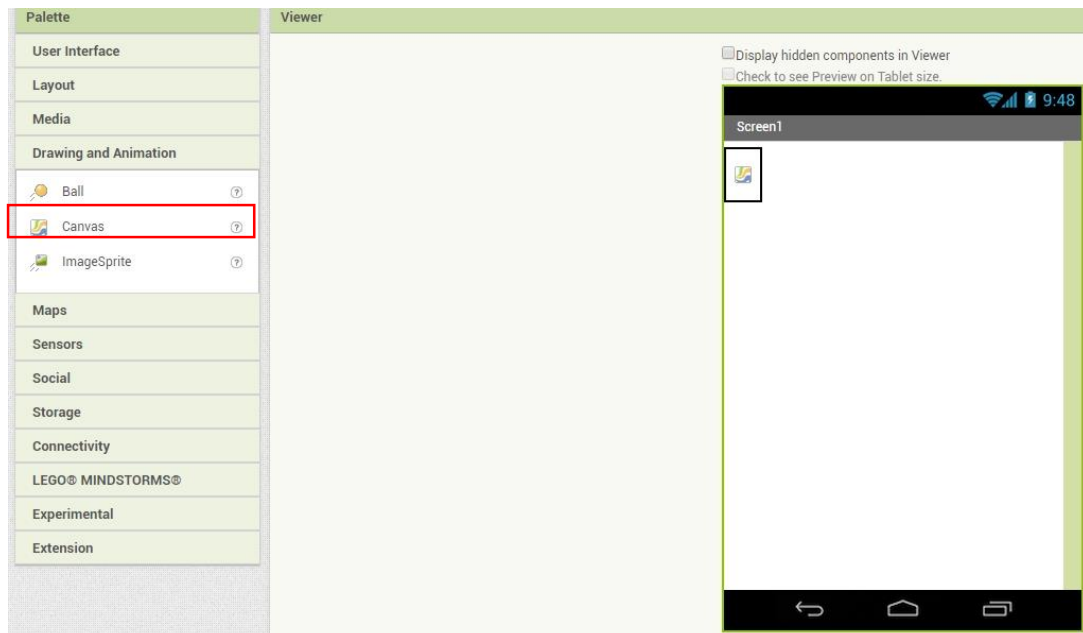
Touched: 可以回傳使用者點擊畫布後的 XY 位置。(點擊後會傳回那個物件的位置)，touchedAnySprite 為 true，代表處碰到動畫元件。



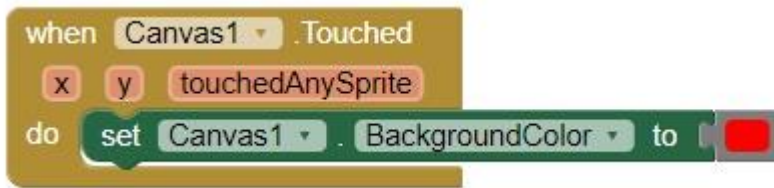
## 範例說明

現在來做一個簡單的範例。

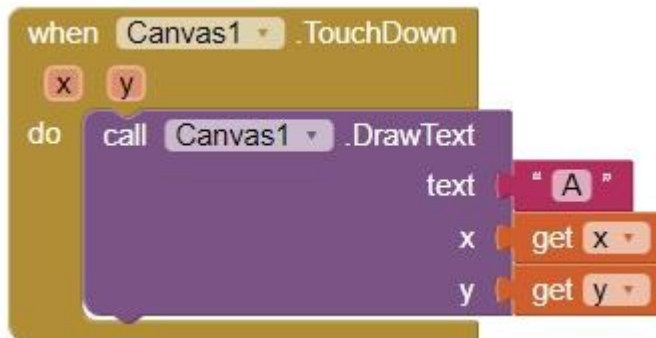
先從旁邊工具列拉出一個畫布。



接著是程式區塊，這邊的意思是，當你點擊畫布時，會將畫布顏色變為紅色。



再來，利用畫布元件中被壓下時會觸發事件，而這裏我們讓他觸發的事件是，會在你畫布的座標那，寫出一個 A 字，所以當這 2 個程式區塊合併時，就會看到一個呈現紅色的畫布，而畫布裡有個英文 A 字。



### 活動

利用上面元件，在畫布上顯示出一個 A 字，顏色自選，並使他可以旋轉與移動，再利用 Save as 將畫布擷取。

## 圖片精靈及球形精靈元件(ImageSprite)

圖片精靈案球形精靈屬於繪圖動畫的元件，他是為動畫和遊戲量身打造的元件，必須與畫布元件配合。

### 圖片精靈及球形精靈元件介紹

圖片精靈通常以圖片當背景圖，並設定方向、速度、旋轉。當圖片精靈碰撞到畫布的邊結實，會觸發到達邊界事件，使用反彈能將其反彈。

球形精靈和圖片精靈相似，差別只在球形精靈只能以半徑設定其半徑大小、畫筆顏色設定單一顏色背景。

TouchDown: 使用者壓住 ImageSprite(圖片精靈或是球形精靈)時並固定不動時，可以回傳使用者壓住的 ImageSprite 的 XY 座標。



TouchUp: 使用者點擊 ImageSprite(圖片精靈或是球形精靈)後手指移開，會回傳使用者手指移開時的 XY 座標。



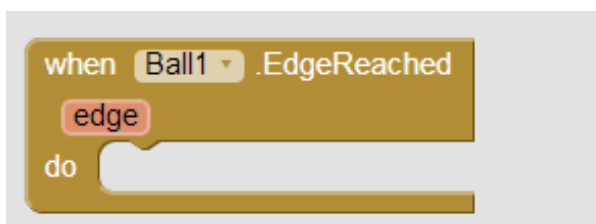
Touched: 當 ImageSprite 被點擊到的時候會觸發此方塊。(作用和 TouchDown 相似；差別在於 TouchDown 是長壓)



NoLongerCollidingWith: 當兩個動畫元件結束碰撞，不再移動時會觸發。



Edge: 將圖片精靈或球形精靈從邊緣一邊元數值方向反彈。例如球彈到邊界時會處發此元件並反彈回來。



MoveIntoBounds: 當 ImageSprite 跑出邊界時，可利用此方塊把他抓回來。例如把球控制在一定框架內反彈。

call ImageSprite1 .MoveIntoBounds

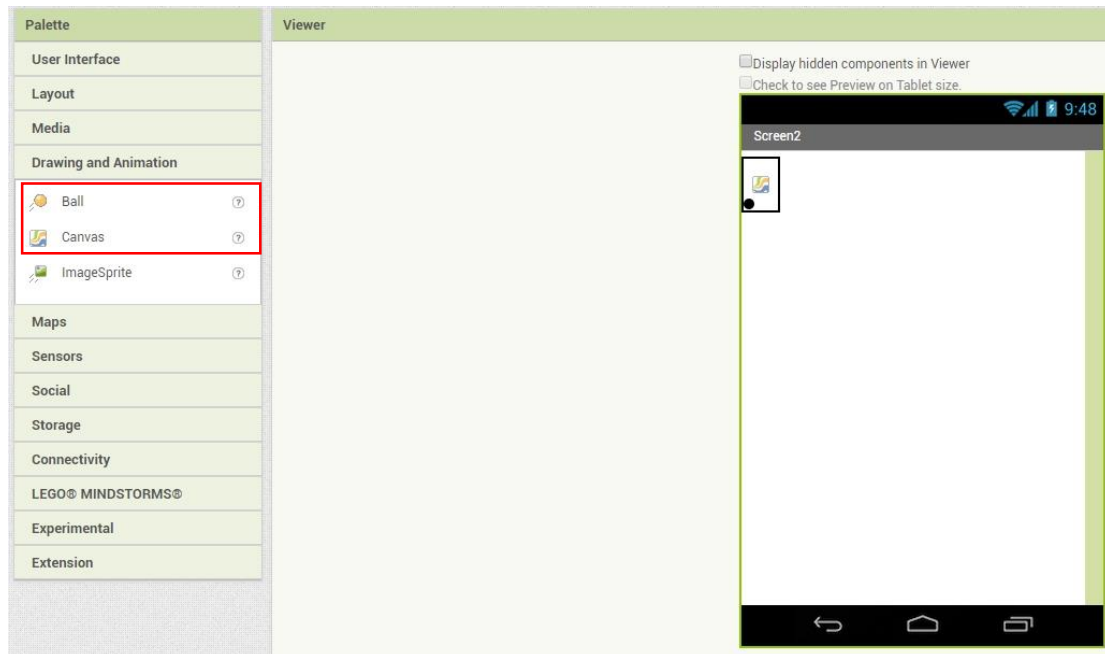
PointInDirection: 讓 ImageSprite 轉向對準指定的 xy 座標方向。例如將球的反彈方向一直控制你想指定的位置。

call ImageSprite1 .PointInDirection  
x  
y

## 範例說明

現在來做個簡單的範例。

從旁邊工具列拉出畫布及球型精靈(注意!球型精靈與圖片精靈只能在畫布裡執行)



再來是程式區塊，這邊的意思是，當球型精靈被按壓時，他便會跑到你設置的座標位置，這邊我們把他的座標位置都設為 10，所以當你按壓他時，會看到它移動到 x，y 座標都會移動 10 的位置。

when Ball1 .TouchDown  
do call Ball1 .MoveTo  
x 10  
y 10

## 活動

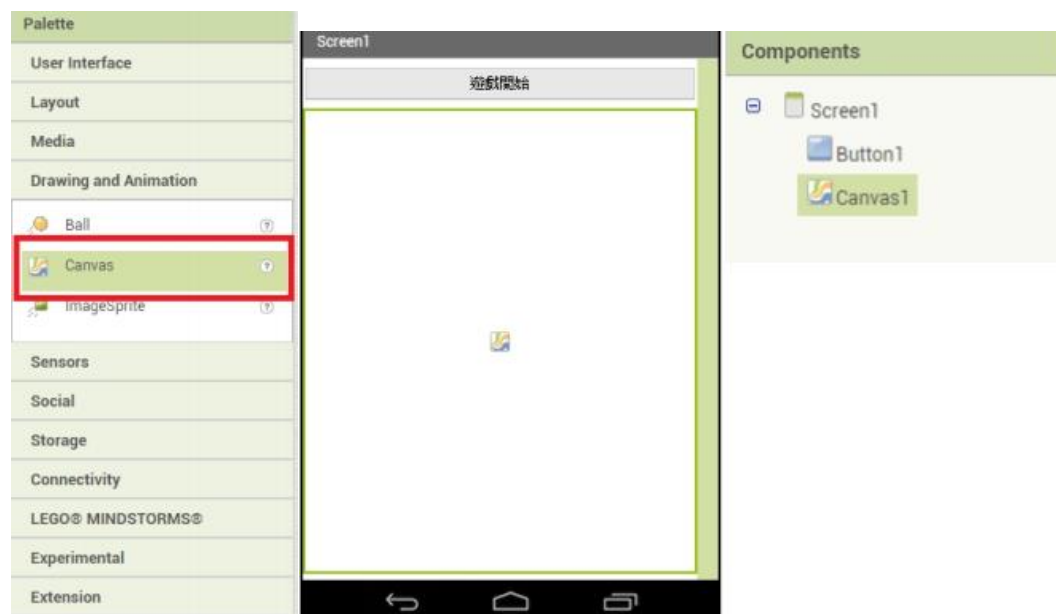
利用上面元件做出 2 個球型精靈，一個是點擊可以回傳座標，一個是長壓回傳座標，並將球控制在一定框架內移動。

## 綜合範例

試著建立一個反彈球的遊戲，當按下遊戲開始按鈕時，球會開始亂跑，碰到邊界便會反彈。

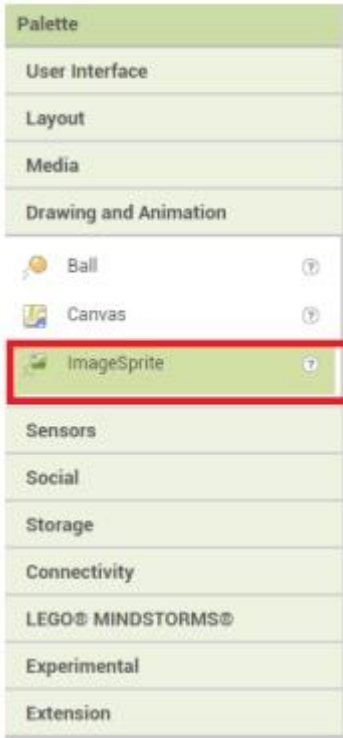
步驟一.

建立一個按鈕 Button(Width 設置成 fill parent)名為遊戲開始，然後拉一個畫布 Canvas 到下面((Height 和 Width 都設成 fill parent)。



步驟二.

接下來拉一個 ImageSprite，位置隨意擺放。



步驟三.

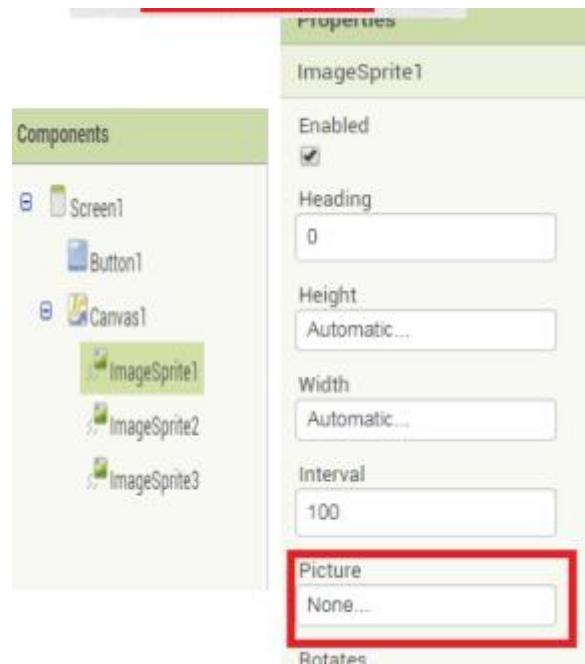
將素材上傳並貼到 ImageSprite

(1) 點擊 Upload File，選擇要上傳的檔案點下 OK。





(2)選擇你要貼圖片的 ImageSprite 點擊右邊的 Picture·然後 點選素材·按下 OK·  
素材就會貼到 ImageSpite 上。

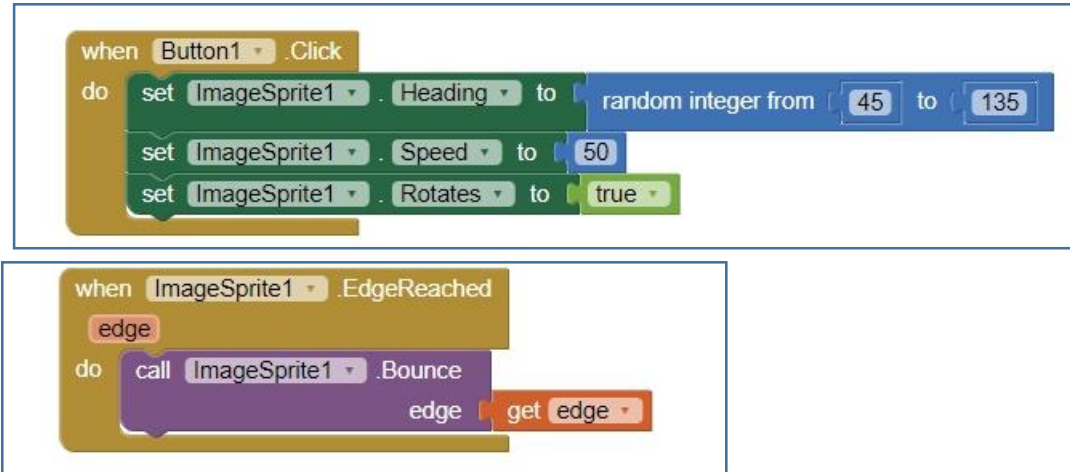


在這裡附上素材圖片



步驟四.

接下來是球的程式區塊。

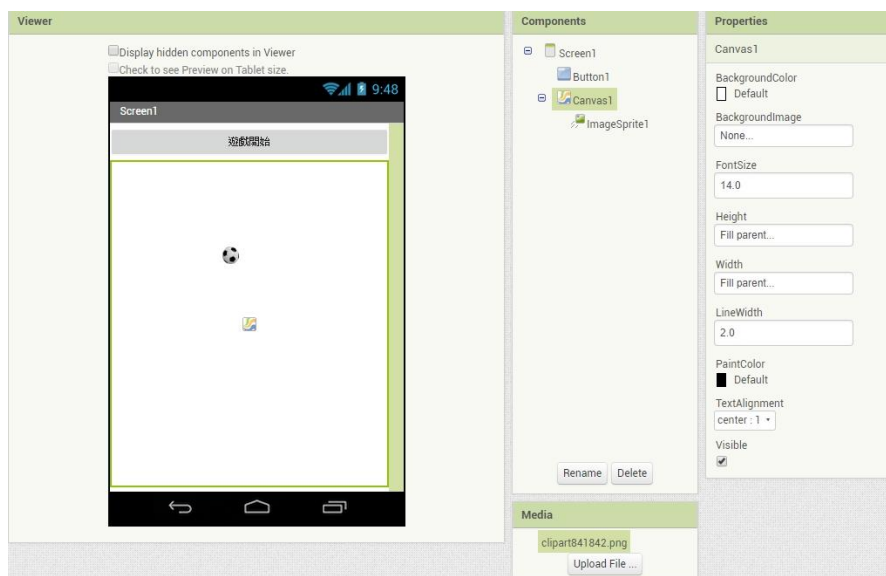


上面的框框是控制球的移動方向、速度，而 `rotates` 設定為 `true`，`ImageSprite` 元件的圖片的 `image` 會旋轉直到對齊 `sprite` 的 `heading` 屬性為止。這邊將球的方向控制在 45 度到 135 度之間，而速度將他調整為 50。

而下面的框框則是畫布邊界的反彈，當球彈到畫布邊框時，會觸發 `edge` 事件，而球也會接收到一個 `get edge` 指令，將其反彈。

步驟五.

開啟模擬器並按下遊戲開始，就能看到球的移動與反彈。



## 練習題

1. 設置一個飛機的圖片精靈，用觸控的方式去移動飛機。
2. 設置兩個圖片精靈，一個是球一個是飛機，讓飛機去觸碰到球能得到分數。
3. 使用下列這 3 張貓咪素材，結合成貓咪在畫布上奔跑的動畫。



4. 在畫布上做出各顏色畫筆，利用觸控的方式去繪圖。
5. 做出一個觸球遊戲，球隨機在邊框反彈，用觸控方式去點擊球，點到球便能加分。