



敬愛的老師您好：

PDF 檔僅限學校教師搭配紙本教材用於  
課堂教學，並未授權其他用途！！

優質好書 · 盡收眼底

 **台科大圖書**

since 1997 用心出版每一本好書



[tkdbooks.com](http://tkdbooks.com)

互動教學 · 活潑有趣

 **hoodaa**  
呼 答 網



<http://hoodaa.net/>

教材題庫 · 及時測評



**MOSME** 行動學習一點通  
Mobile Online Study Made Easy.



[mosme.net](http://mosme.net)

線上學習 · 沒有距離



**優勢學習網**  
[www.use7.net](http://www.use7.net)



<http://use7.net/>

書籍配套 · 軟硬兼俱

 **紅動創新**



<http://ipoemaker.com/>

研習競賽 · 沒有問題

**iPOE 活動網**  
Intelligent · Public · Open · Easy-learning Event Platform



<http://pacme.asia/>

前進專業 · 淬煉身價

**勁園科技教育**

Jin Yuan TechEdu



<http://jytechedu.com/>

科技教育 · 先覺先知

**iPOE 國際學院**  
Intelligent · Public · Open · Easy-learning International College



<http://ipoe.cc/>

深入淺出

# 用 MakeCode Arcade

## 玩程式設計第一次就上手

### 使用 WiFiBoy Arcade 編程學習遊戲機

康文耀 編著



程式檔案下載說明：

本書程式檔案請至台科大圖書網站 ([tkdbooks.com](http://tkdbooks.com)) 圖書專區下載；或可直接於台科大圖書網站首頁，搜尋本書相關字（書號、書名、作者），進行書籍搜尋，搜尋該書後，即可下載本書程式檔案內容。



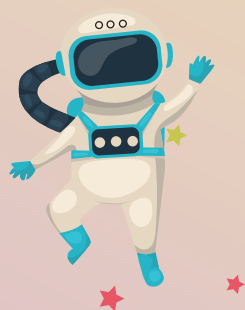
# 序

筆者從事教職工作歷經職業學校資料處理科、資訊科直至赴國中擔任資訊科技教師，投身程式設計軟、硬體教學實務近二十年。對於程式教育有著一股熱忱，總是希望學生學習不是為了應付老師，而是能在學習過程中發現樂趣、了解原來這些神奇的事物並沒有想像中的困難，也期望透過本書讓更多的程式初學者能有個好的入門磚。

MakeCode Arcade 是 2019 年 1 月 18 日由微軟公司所發布，內含 2D 遊戲引擎，比 Scratch 更簡單、有趣、好玩，且支援積木式和文字式程式語言，非常適合初學者及專業玩家學習的開源軟體。MakeCode Arcade 可以輕易的設計出復古街機遊戲運行於網頁和小型遊戲機，網頁部分無需安裝，遊戲就能直接在電腦、平板、手機的瀏覽器中設計和執行；也可將設計好的程式下載到實體遊戲機，也就是本書推薦採用的 WiFiBoy Arcade 編程學習遊戲機。此遊戲機內建擴充母座不需要像 Scratch 還要安裝中介軟體，便可直接連接外部感測器或控制外部設備，使用上非常簡單又方便。

筆者才疏學淺，本書如有疏漏謬誤之處勢必在所難免，懇請教育先進們海涵並不吝給予指正。本書得以順利付梓，得賴 WiFiBoy Arcade 學習型遊戲機研發者賴溱君及楊敏瀟二位先生的協助並提供軟、硬體支援，慧燈中學楊岱霖老師協助本書審校，尤其感謝台科大圖書王書中經理和編輯團隊們的辛勤付出。

康文耀 謹誌



# 目錄



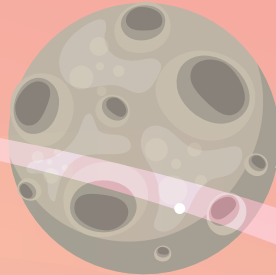
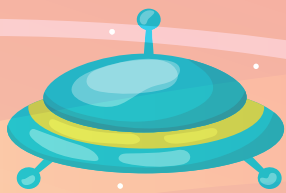
## 認識 WiFiBoy Arcade 與 MakeCode Arcade

1.1	WiFiBoy Arcade 學習型遊戲機	2
1.2	下載程式到遊戲機	3
1.3	MakeCode Arcade 編輯器介紹	6
1.4	MakeCode Arcade 說明文件	7
1.5	開啟積木做的遊戲範例程式	9
1.6	編譯並下載程式至 WiFiBoy Arcade 遊戲機	12
1.7	試做一個自己設計的遊戲	14
1.8	積木編輯技巧	19
1.9	儲存專案和分享	20
1.10	重置或刪除專案	22



## Arcade 遊戲設計概念

2.1	事件積木	24
2.2	遊戲背景	24
2.3	遊戲角色與角色移動	29
2.4	偵測角色類別重疊	33
2.5	遊戲聲音和畫面特效	34
2.6	更換背景貼圖	38
2.7	結束遊戲	41
2.8	完成的積木程式碼	44
2.9	在程式編輯區中常用的按鍵	45
	綜合實作題	46



## 修改「太空驅逐艦」 (Space Destroyer) 範例

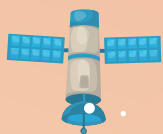
3.1	開啟積木做的遊戲範例程式	48
3.2	事件積木	48
3.3	「當啟動時」事件積木（遊戲初始設定）	49
3.4	「當 A 鍵被按住」事件積木（按 A 鍵發射砲彈）	57
3.5	「當遊戲更新間隔」事件積木（每 0.5 秒掉落隕石）	60
3.6	「碰撞偵測」事件積木（偵測太空船和隕石碰撞）	62
3.7	「碰撞偵測」事件積木（偵測砲彈和隕石碰撞）	64
3.8	本章使用到 Arcade 遊戲設計相關名詞	66
3.9	修改後完整程式碼	67



## 「收集隕石任務」遊戲

4.1	遊戲設計分析	70
4.2	「當啟動時」事件	71
4.3	「重複無限次」事件	72
4.4	「當遊戲更新」事件	73
4.5	「角色類別重疊」事件	74
4.6	「角色類別重疊貼圖」事件	74
4.7	完整程式碼	75
	綜合實作題	76





# 5 Chapter

## 「離線小鴨鴨」遊戲

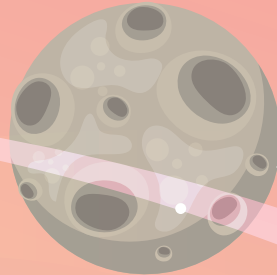
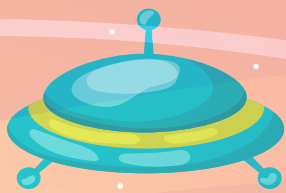
5.1	遊戲設計分析	78
5.2	「當啟動時」事件	79
5.3	「當任何鍵被按住時」事件	81
5.4	「當遊戲更新」事件	81
5.5	「當遊戲更新 1000 毫秒」事件	82
5.6	「重複無限次」事件	83
5.7	「當類別 Player 重疊類別 Projectile」事件	83
5.8	完整程式碼	84

# 6 Chapter

## 「小鴨過街」遊戲

6.1	遊戲設計分析	86
6.2	「當啟動時」事件	88
6.3	當方向按鈕「按下」事件	89
6.4	「當遊戲更新 500 毫秒」事件	90
6.5	「當遊戲更新 1000 毫秒」事件	91
6.6	「當類別 Player 重疊類別 Enemy」事件	92
6.7	「當類別 Player 重疊類別 Home」事件	92
6.8	完整程式碼	93





## 「射擊雙打」遊戲

7.1	遊戲設計分析	96
7.2	「當啟動時」事件	97
7.3	「當玩家 1 按下 A 鍵時」事件	98
7.4	「當玩家 2 按下 A 鍵時」事件	99
7.5	「當遊戲更新間隔 500 毫秒」事件	100
7.6	「當玩家類別重疊飛碟類別」事件	101
7.7	「當子彈類別重疊飛碟類別」事件	102
7.8	完整程式碼	103



## 「大樂透開獎」遊戲

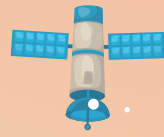
8.1	遊戲設計分析	106
8.2	「當啟動時」事件	107
8.3	「當按下 A 鍵時」事件	109
8.4	完整程式碼	111



## 「指尖跳舞」遊戲

9.1	遊戲設計分析	114
9.2	「當啟動時」事件	115
9.3	「重複無限次」事件	118
9.4	「當方向鍵被按住時」事件	119
9.5	「當角色類別重疊時」事件	120
9.6	完整程式碼	123





# 10 Chapter

## 自訂 Arcade 文字積木

10.1 使用積木程式自訂積木指令	126
10.2 撰寫 JavaScript 自訂文字積木指令	128
10.3 申請 GitHub 帳號	130
10.4 製作自訂積木文件	134
10.5 上傳自訂積木文件	137
10.6 使用擴展自訂積木	140

# 11 Chapter

## 「電子輪盤」遊戲

11.1 遊戲設計分析	144
11.2 「當啟動時」事件	145
11.3 「當 A 鍵被按住」事件	147
11.4 「當 B 鍵被按住」事件	148
11.5 完整繁體中文程式碼 綜合實作題	149 150

# 附錄

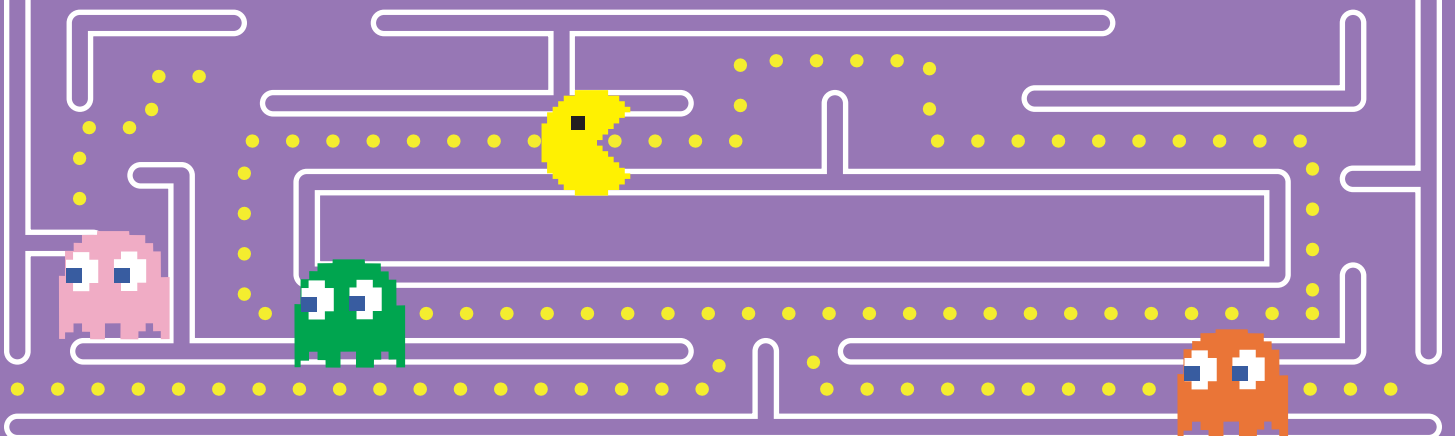
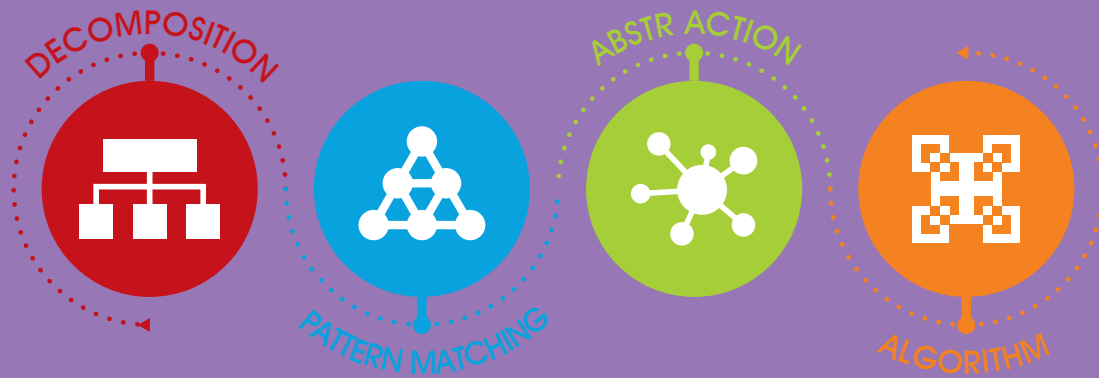
附錄一、各章節練習參考解答	152
附錄二、Arcade 常用積木指令範例	159
附錄三、Arcade 積木指令介紹	187



# 認識 WiFiBoy Arcade 與 MakeCode Arcade

想要做遊戲的初學者，該學習哪一種程式語言呢？推薦使用 MakeCode Arcade 積木來製作遊戲，能在做遊戲的過程中學習運算思維。

本章介紹如何使用 MakeCode Arcade 積木製作遊戲，完整的設計範例將針對遊戲框架與邏輯概念、主角與遊戲設定、運算應用與遊戲樂趣來訓練基礎邏輯、思辨能力與問題解決能力。

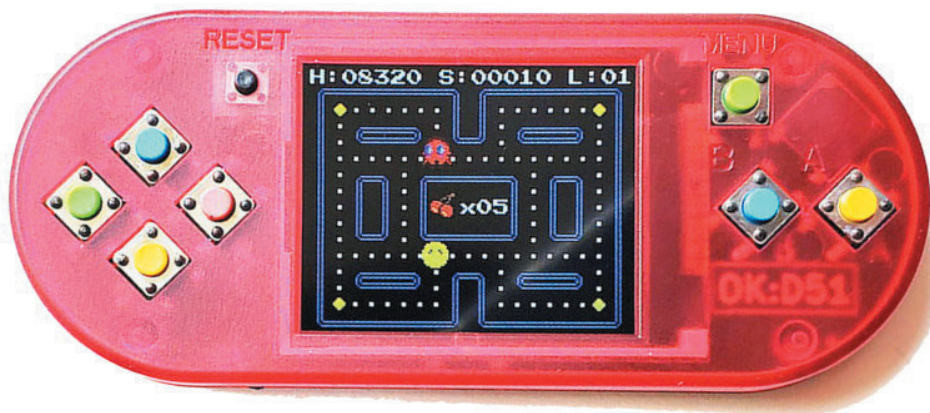


## 新手村引導

學習任一種全新的知識，入門的學習最為關鍵，有了好的開始就能持續的學習，因此，擁有一台學習型遊戲機，用來提升學習興趣是學習做遊戲的最佳利器。當然，使用 MakeCode Arcade 線上模擬器，也能完整的學習做遊戲和訓練基礎邏輯與思辨能力，讓我們開始吧。

## 1.1 WiFiBoy Arcade 學習型遊戲機

WiFiBoy Arcade 是一部讓國小到高中的學生，都能玩運算思維與訓練程式設計能力的學習型遊戲機，支援使用 MakeCode Arcade、MicroPython、JavaScript、Arduino 等程式語言編輯器來製作遊戲的驗證平台，本書藉由程式積木及遊戲機的結合應用，在 Microsoft MakeCode Arcade 編輯器上所設計、製作的遊戲，馬上就可以下載到 WiFiBoy Arcade 遊戲機上運作，讓學生都能做個自己的遊戲，提升學生學習興趣，快快樂樂增強邏輯思考力！



 WiFiBoy Arcade

WiFiBoy Arcade 左側有四個控制方向的按鍵和一個重置（RESET）按鍵，中間是一個 1.8 吋彩色 TFT 液晶螢幕，解析度為 160×128 點，右側上面是選單（MENU）按鍵和遊戲按鍵 A 和按鍵 B，內部有發聲器、三軸加速度感應器和 12 腳位的擴充母座，擴充母座可連接外部感測器或控制外部設備，也能將二台 WiFiBoy Arcade 連線設計對打遊戲。

WiFiBoy Arcade 官方網站有許多相關資源可下載 <https://wifiboy.org/arcade/>

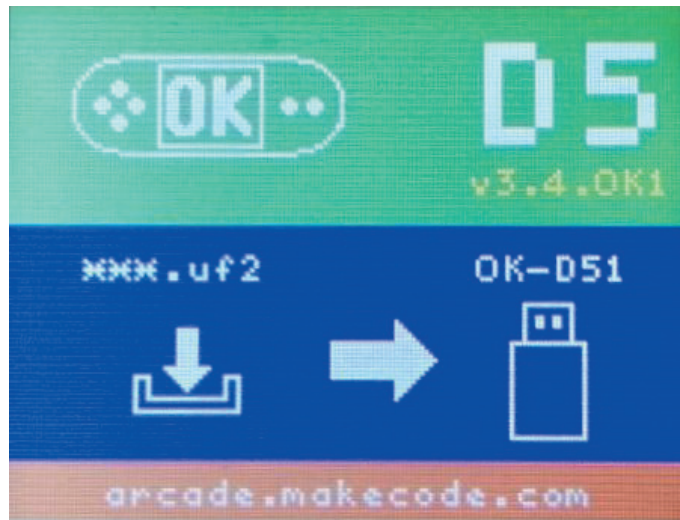
## 1.2 下載程式到遊戲機

當 WiFiBoy Arcade 使用 USB 訊號傳輸線連接至電腦後，會在電腦上自動產生「OK-D51」虛擬磁碟機，有兩種方式可將遊戲程式下載至遊戲機中，第一種是使用已經編譯好的程式（.uf2），將 .uf2 檔案複製到「OK-D51」虛擬磁碟機中即可；第二種是透過後面介紹的 MakeCode Arcade 編輯器，將設計好的遊戲程式編譯並下載到遊戲機中，以下先介紹第一種下載方式：

### 一、安裝「小精靈」遊戲

**step 1** 將 WiFiBoy Arcade 遊戲機接上 MicroUSB 訊號傳輸線到電腦。

**step 2** 按遊戲機重置（RESET）鈕一次（或二次），使遊戲機畫面出現下載程式畫面，如下圖：



⦿ 下載程式畫面

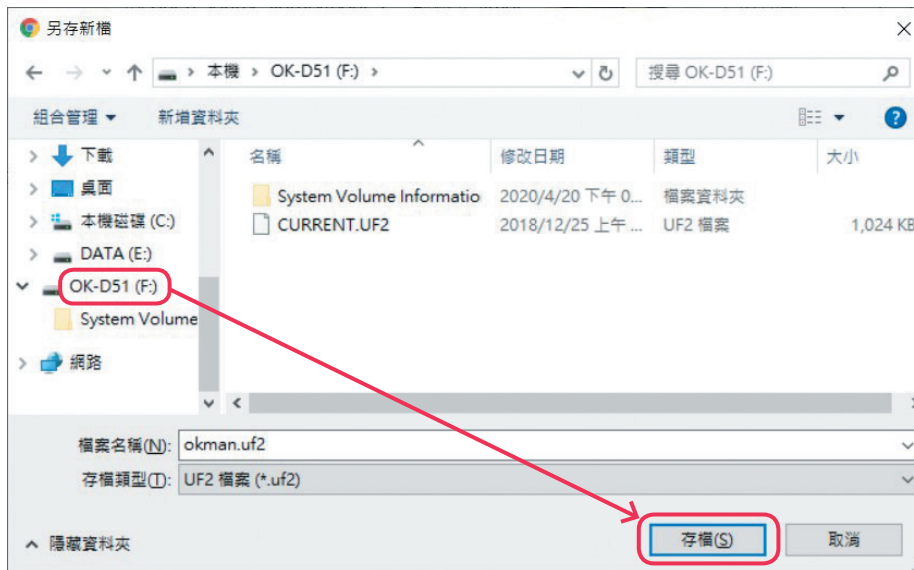
**step 3** 此時，電腦會自動新增一個「OK-D51」虛擬磁碟機。



⦿ 自動新增一個「OK-D51」虛擬磁碟機

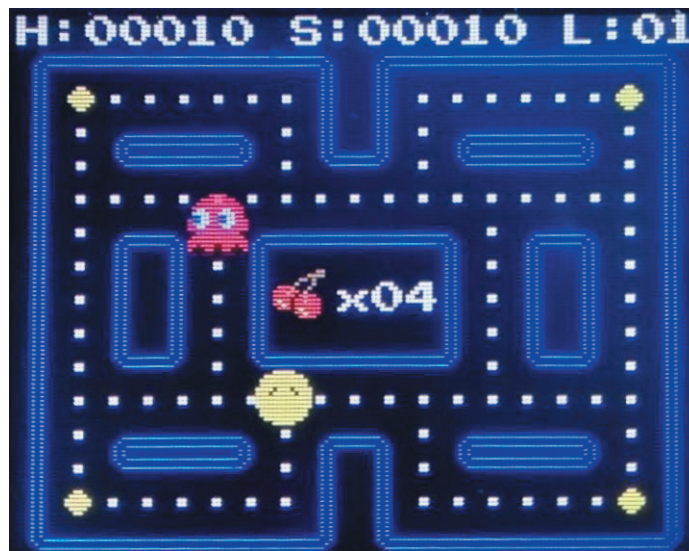
**step 4** 電腦開啟瀏覽器，到網址 <https://wifiboy.org/arcade/files/okman.uf2> 下載已編譯好的小精靈遊戲（okman）檔案。

**step 5** 將 .uf2 檔案另存至「OK-D51」虛擬磁碟機。



 另存至「OK-D51」虛擬磁碟機

**step 6** 下載完成後就能在遊戲機上玩小精靈遊戲（okman）。



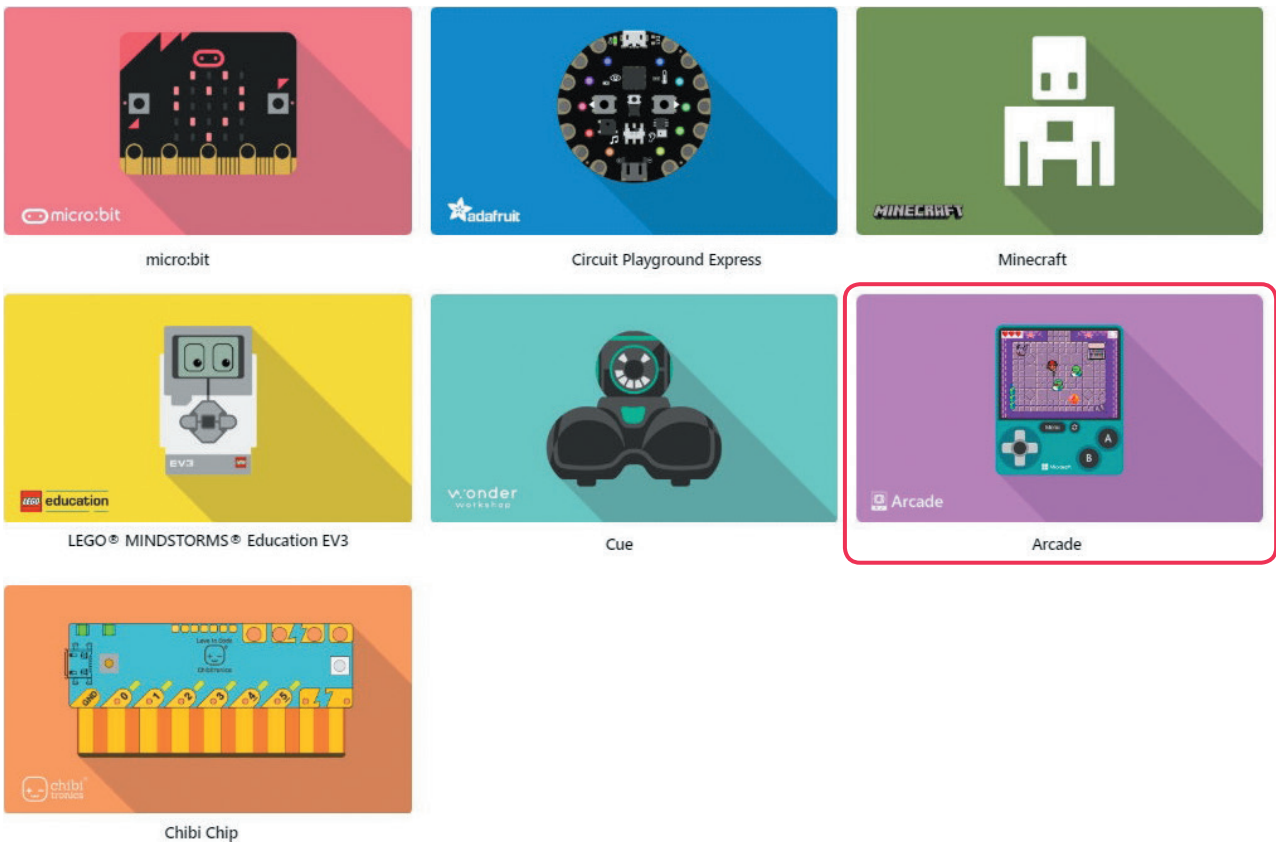
 okman 畫面

## 二、Microsoft MakeCode Arcade 編輯器

Microsoft MakeCode 是微軟開發的免費開放原始碼平台包括許多有趣專案，互動式模擬器可模擬即時執行結果，在程式執行時提供即時回饋，方便測試程式碼並進行偵錯，適合不同學習程度的積木編輯器和文字編輯器，WiFiBoy Arcade 使用的是其中的 Arcade 編輯器，Arcade 中文譯為大型電玩，俗稱街機。

### 小精靈引導



Microsoft MakeCode 網址 <https://makecode.com/>

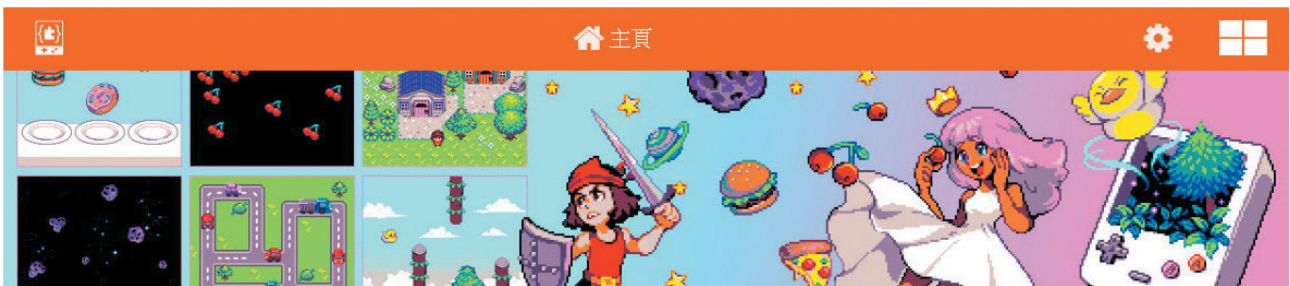


Microsoft MakeCode 各種編輯器

## 1.3 MakeCode Arcade 編輯器介紹

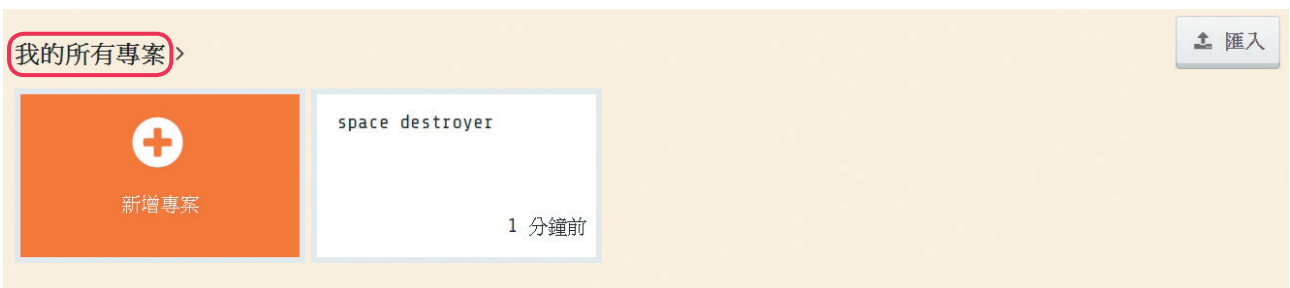
製作遊戲最重要的開發工具是「遊戲引擎」，MakeCode Arcade 編輯器內含超好入手的遊戲引擎，透過這個遊戲引擎內建的碰撞偵測系統、音效、動畫以及場景管理等，很容易就可設計出經典的街頭電玩遊戲，如小蜜蜂、打磚塊、太空侵略者……等遊戲。

瀏覽器輸入 MakeCode Arcade 編輯器線上版網址：<http://arcade.makecode.com/>，進入 MakeCode Arcade 編輯器的「主頁」，首先看到的是一個主頁橫幅，橫幅左上角圖示  為 MakeCode Arcade 軟、硬體相關文件的連結，右上角齒輪圖示  為「設定語言」、「開啟高對比顯示」（黑色背景）及「重置」（清除所有專案）設定。



◎ 主頁橫幅

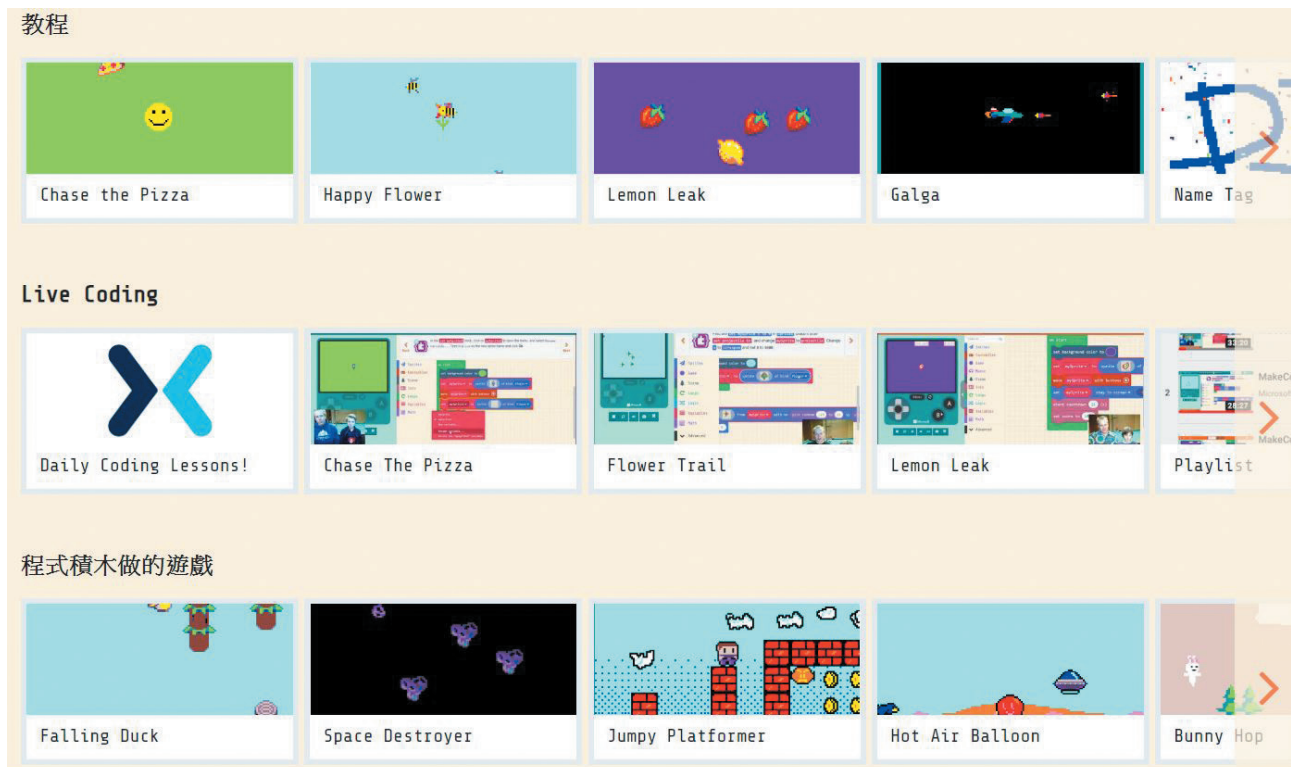
主頁橫幅下方是最近使用過的「專案」列表，每一個遊戲程式稱為一個專案，在 MakeCode 中創建的專案將保存到瀏覽器所在的位置，按下「我的所有專案」可顯示在這台電腦中曾經開啟過的專案。



◎ 我的所有專案


上圖右側的「匯入」鈕，可匯入未曾開啟的遊戲專案，左側「新增專案」則可新增一個空白的遊戲程式，新增專案時如果沒有給專案命名，則編輯器會自動命名為“未命名”保留在電腦中。

主頁下面為許多教學影片及遊戲範例連結，其中「教程」區塊中的連結，可以讓初學者依照指示，一步一步完成一個簡單的遊戲，並熟悉編輯器；「Live Coding」區塊中的連結，為一對父子利用互動對話方式教導初學者的教學影片；「程式積木做的遊戲」區塊中的連結，是一些完整的遊戲範例程式，初學者可先從修改範例程式開始學習。



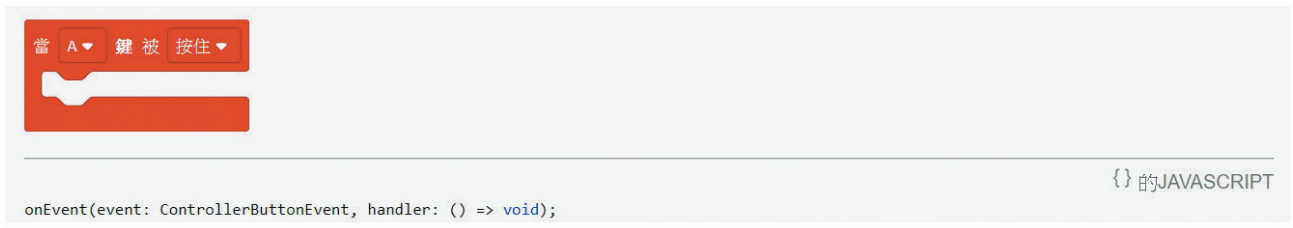
🕒 教學及範例連結

## 1.4 MakeCode Arcade 說明文件

MakeCode Arcade 說明文件網頁：<https://arcade.makecode.com/docs>，在說明文件網頁左上角圖示  為說明文件選單，點選子選項後，右側網頁會顯示對應的相關說明，在修改範例程式時，遇到不容易理解的積木，都能在這裡找到解答，點選積木後還有詳細說明和操作範例。



[MakeCode Arcade 說明文件](#)

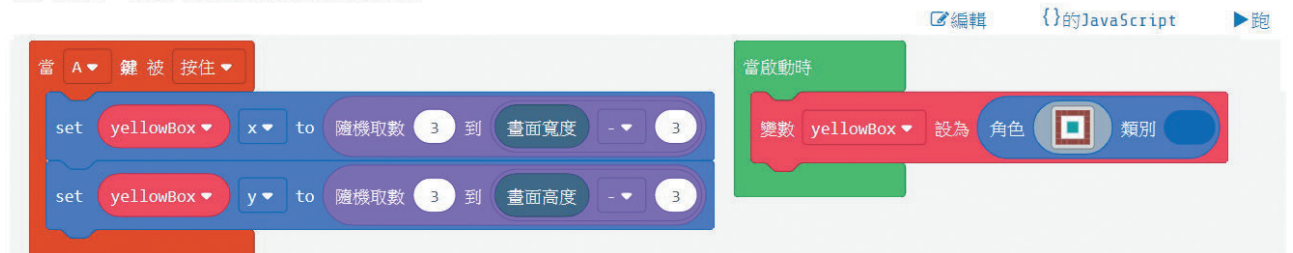


### 參量

- **event** : 要等待的按鈕動作。按鈕動作 (事件) 是 :
  - `pressed` : 按下按鈕
  - `released` : 按鈕被釋放
  - `repeated` : 按下按鈕後, 此事件會不斷重複
- **handler** : 當按鈕發生故障時要運行的代碼
  - `repeated` : 按下按鈕後, 此事件會不斷重複

### 例

按下按鈕時, 將黃色框移動到屏幕上的隨機位置。



[詳細說明和操作範例 \(使用 Chrome 瀏覽器自動翻譯\)](#)

## 1.5 開啟積木做的遊戲範例程式

在 MakeCode Arcade 編輯器「主頁」點選「程式積木做的遊戲」區塊中的太空驅逐艦積木遊戲「Space Destroyer」。



⦿ 「太空驅逐艦」積木遊戲

畫面顯示遊戲 GIF 動畫短片，點選「打開範例」。



⦿ 打開「太空驅逐艦」範例

打開「太空驅逐艦」範例後，進入遊戲編輯畫面，MakeCode Arcade 編輯畫面主要分成「模擬器」、「積木指令區」和「程式編輯區」三區：



⦿ MakeCode Arcade 遊戲編輯畫面

## A. 模擬器

可立即驗證遊戲設計成果，模擬器的右側「<」圖示可將模擬器隱藏至畫面左側，讓程式編輯區空間更大。

「切換偵錯模式」可啟用單步執行，很容易找出有問題的程式。

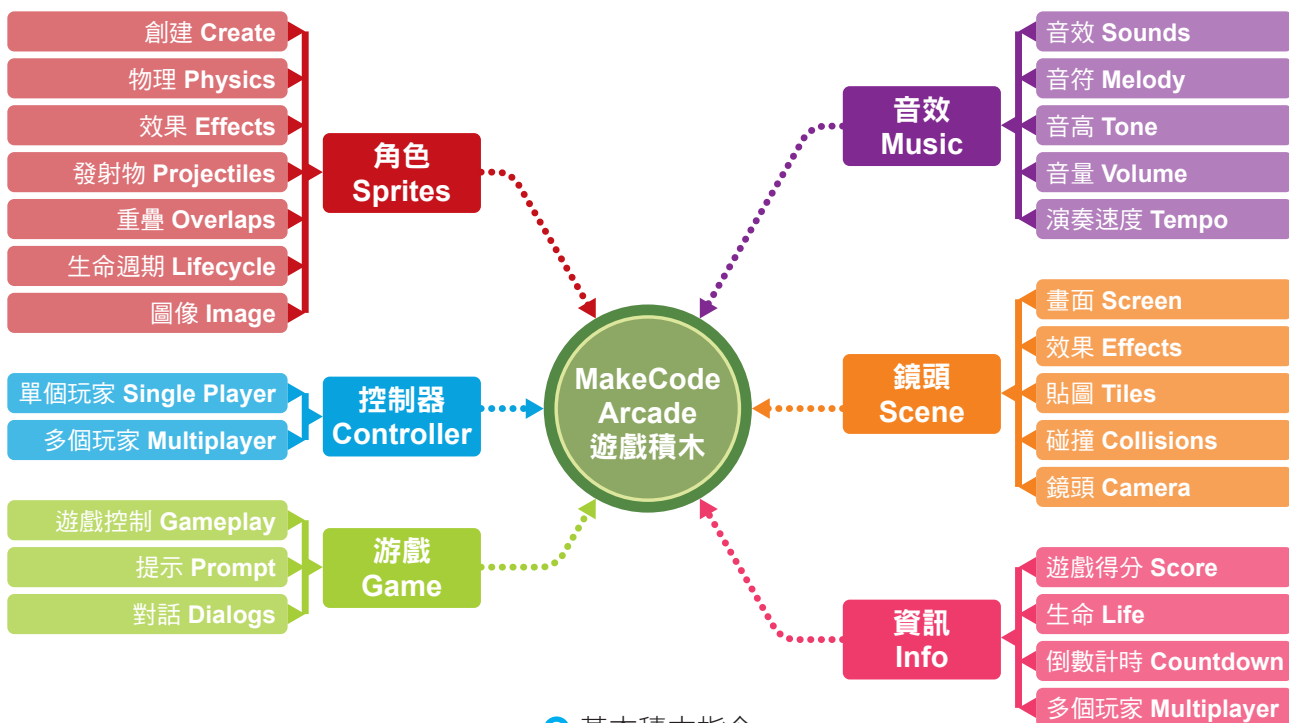
「瀏覽操作按鍵」顯示使用鍵盤玩遊戲時對應的按鍵，使用電腦鍵盤在模擬器上玩遊戲需將滑鼠滑過模擬器上方，使模擬器上的控制器呈現高亮度狀態。

「擷取遊戲畫面」可將遊戲存成 PNG 檔（Perfect Nice Game 模式），之後可使用主頁中的「匯入」功能，將積木遊戲程式匯入到其他電腦繼續設計。



 模擬器功能鍵

## B. 積木指令區



 基本積木指令

在 Arcade 編輯器中常用到的積木依其指令操作方式可分成以下類型：

類型	積木圖式	操作說明
事件積木		當該事件被觸發時，才會執行事件內的積木，所有積木指令都必須放在事件積木中。 例如：當按鈕 A 被按住時，就會去執行事件積木內的積木程式。 ※ 滑鼠在事件積木上按右鍵，可將事件積木收合起來或添加註解。
控制積木		長條形上凹下凸，可方便與事件、設定和控制積木結合，主要控制遊戲的動作或效果，必須放在事件積木中。 例如：將 mySprite 角色從遊戲中刪除，點開「+」可增加其他功能設定。
設定積木		長條形上凹下凸，可方便與事件、設定和控制積木結合，主要讓右邊的參數內容取代或改變指定的項目內容。 例如：設定讓變數 A 的內容為 0 (A=0)。
布林值積木		外形是六邊形，負責回傳布林運算結果，例如：真 (True) 或假 (False)。一般會放在條件積木中當做某段程式是否執行的依據。 例如：如果變數 A 小於 0 則傳回 true 否則傳回 false。
取值積木		外形是橢圓形，包含數學的運算積木，負責傳回偵測值或運算結果，其結果可做為其他積木的參數。 例如：獲取遊戲機屏幕寬度。
迴圈積木		迴圈是一段在程式中可能會連續執行多次的程式碼，必須放在事件積木中。 例如：重複執行 4 次迴圈積木內的程式碼。
條件積木		條件積木 (if) 的運用，主要是用來判斷設定的條件是否成立，來決定要執行的程式碼區段，必須放在事件積木中，多數情況是使用布林值積木做為判斷依據。

### C. 程式編輯區

積木式程式語言，是將積木指令結合在一塊，然後組合出可以在遊戲機上運行的程式，在程式編輯區中所有的積木指令應該都被置於「事件類型」（例如：當按鍵被按住、當角色重疊…）的積木中，「當啟動時」的事件積木是首先會被執行，並且只會執行一次的事件，程式編輯區內如有未用到的積木（呈灰色），必須拖曳至「積木指令區」中刪除該積木，以避免影響程式的正確執行。

「程式編輯區」的下方操作按鈕說明如下：



◎ 專案儲存按鈕

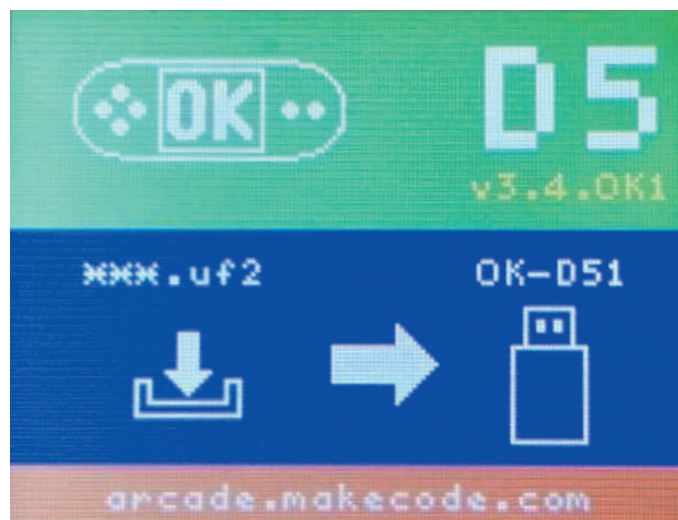
◎ 積木操作按鈕

## 1.6 編譯並下載程式至 WiFiBoy Arcade 遊戲機

要將設計好的積木程式下載至遊戲機時，需將遊戲機和電腦連線，並讓 Arcade 編輯器將程式編譯成副檔名為「.uf2」的檔案，再將編譯好的檔案存入遊戲機中，操作步驟如下：

**step 1** 將 WiFiBoy Arcade 遊戲機接上 MicroUSB 訊號傳輸線到電腦，並開啟遊戲機電源。

**step 2** 遊戲機畫面出現「下載程式」畫面，如沒有出現「下載程式」畫面請按遊戲機重置（RESET）鈕一次（或二次），下載程式畫面如下圖：



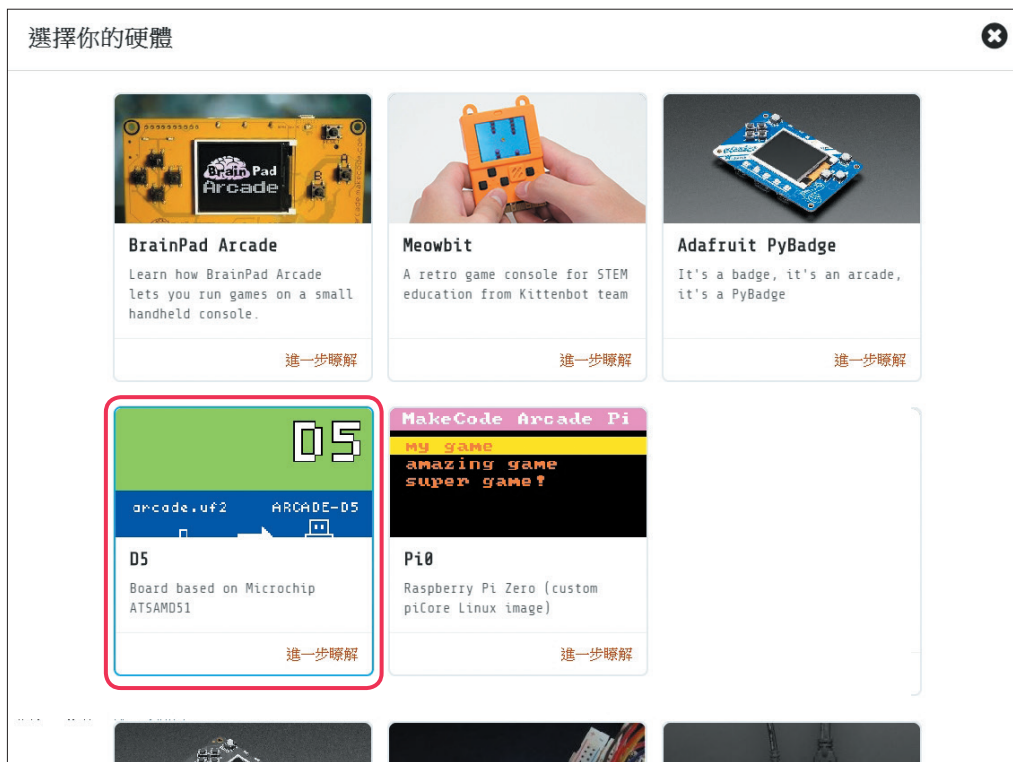
◎ 下載程式畫面

**step 3** 按下 Arcade 模擬器下面「下載」鈕，即開始編譯並準備下載程式至遊戲機中。



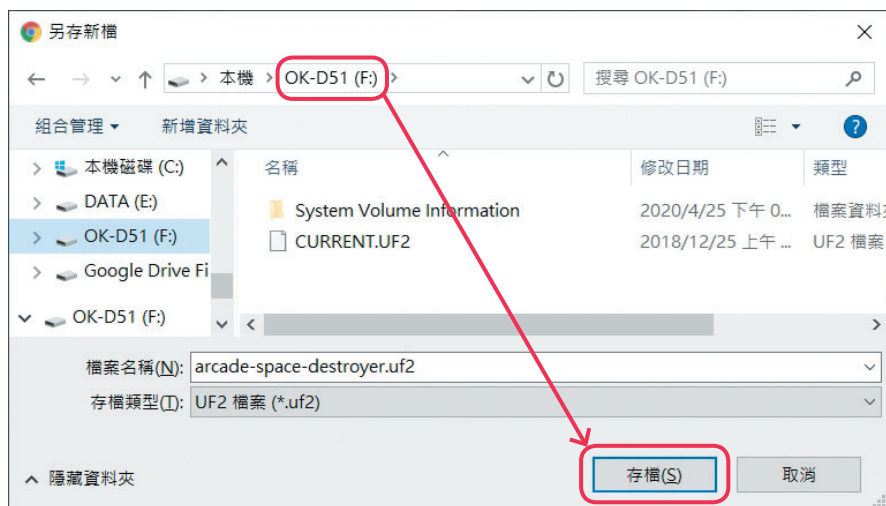
⦿ 下載程式至遊戲機

**step 4** 第一次下載程式至遊戲機時，會出現「選擇你的硬體」視窗，點選「D5」硬體圖示。



⦿ 選擇 D5 硬體

**step 5** 出現「另存新檔」視窗時，選擇「OK-D51」虛擬磁碟機，按下「存檔」鈕即可。



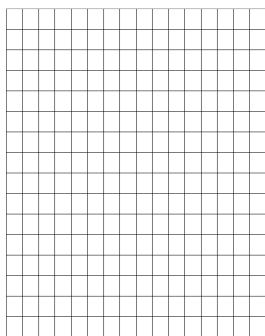
⦿ 下載程式到遊戲機

**step 6** 如果遊戲機前一次的下載程式是用積木程式編譯的，遊戲機按一次重置（RESET）鈕即可進入「下載程式」畫面，如果下載程式是用 Arduino 程式編譯的，則需按二次重置（RESET）鈕才會進入「下載程式」畫面。

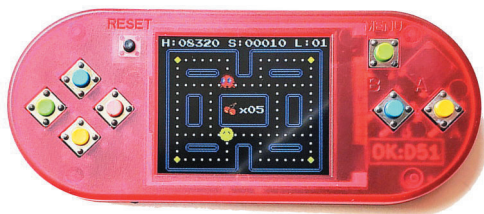
## 1.7 試做一個自己設計的遊戲

要玩自己的程式遊戲要請你先準備如下：

1. 準備一台電腦並開啟瀏覽器上網至 <https://arcade.makecode.com/>。
2. 找一台實體小車子，並在下面的格子中使用這 15 種顏色繪製它。



3. 準備一台 WiFiBoy Arcade 編程學習遊戲機。



#### 4. 依照步驟完成手動操控小車遊戲

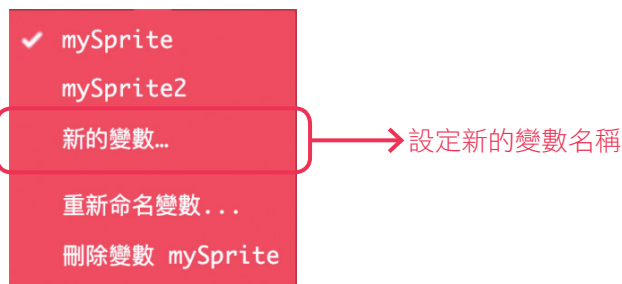
**step 1** 滑鼠左鍵點擊「角色」積木盒中的「變數設為角色」積木指令



**step 2** 將「變數設為角色」積木指令拖曳至「當啟動時」事件積木內



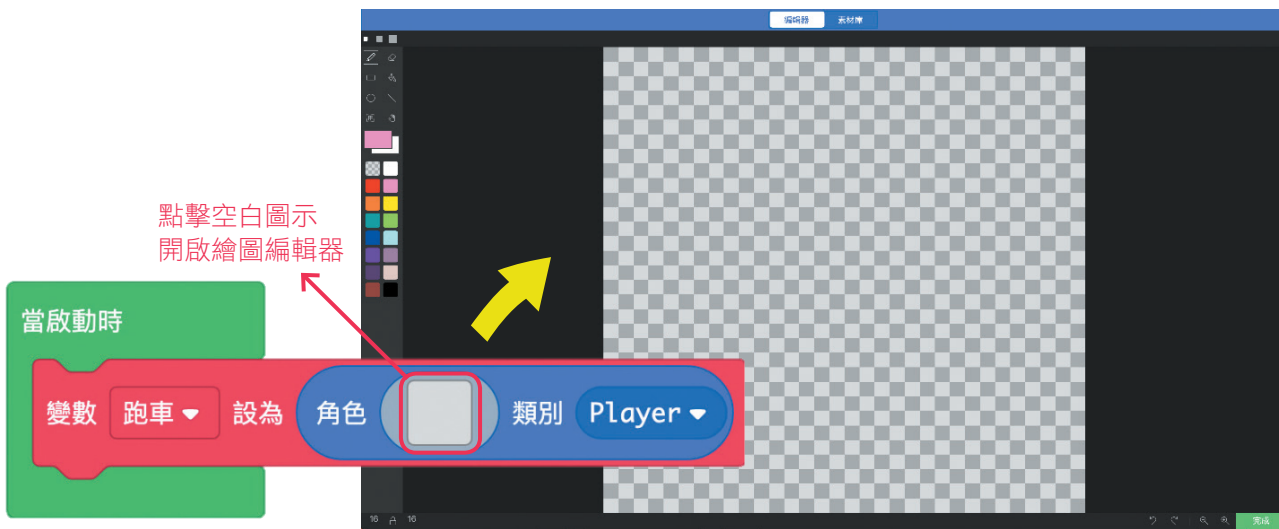
**step 3** 下拉「mySprite」變數名稱，選擇「新的變數」



**step 4** 在「新變數的名稱」視窗中，輸入「跑車」



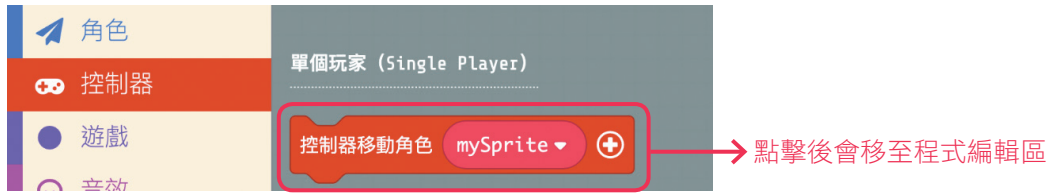
**step 5** 點擊角色空白圖示開啟「繪圖編輯器」



**step 6** 先設定圖形解析度為「18×12」像素，再依照實體小車子開始繪製遊戲角色，繪製完成後按「完成」鈕返回編輯區



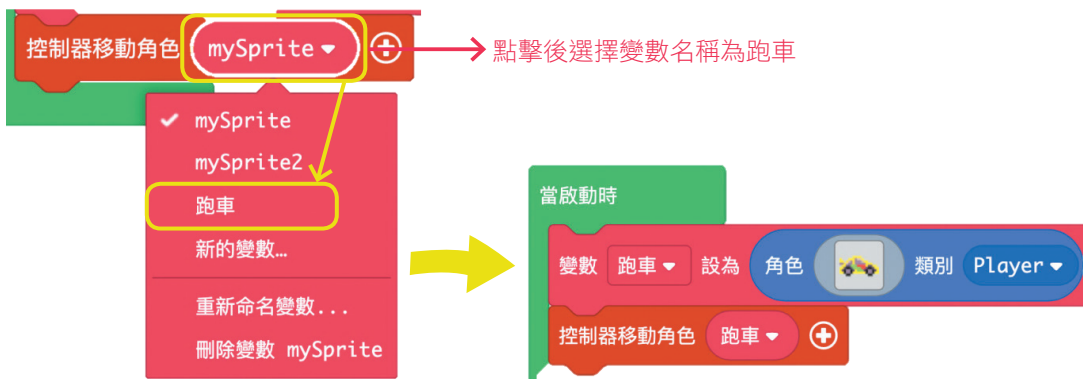
step 7 滑鼠左鍵點擊「控制器」積木盒中的「控制器移動角色」積木指令



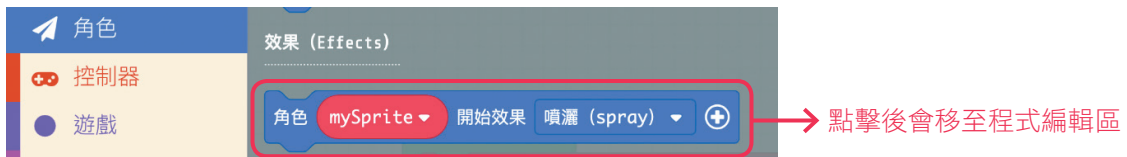
step 8 將「控制器移動角色」積木指令拖曳至「當啟動時」事件積木內



step 9 下拉「mySprite」角色名稱，選擇「跑車」



step 10 滑鼠左鍵點擊「角色」積木盒中的「角色開始效果」積木指令



step 11 將「角色開始效果」積木指令拖曳至「當啟動時」事件積木內



**step 12** 下拉「mySprite」角色名稱，選擇「跑車」，下拉開始效果選擇「暖色放射狀」效果



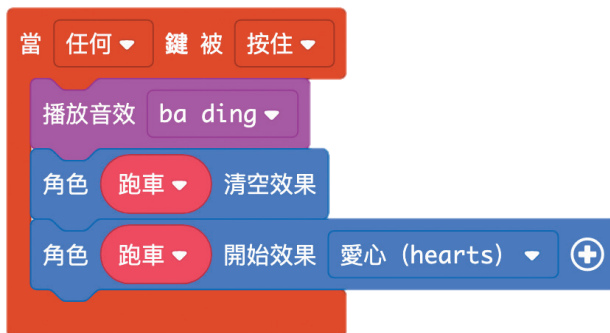
**step 13** 完成後即可使用控制器的方向鍵，控制「跑車」角色在屏幕上自由移動。

### 想一想！可以試著增加功能

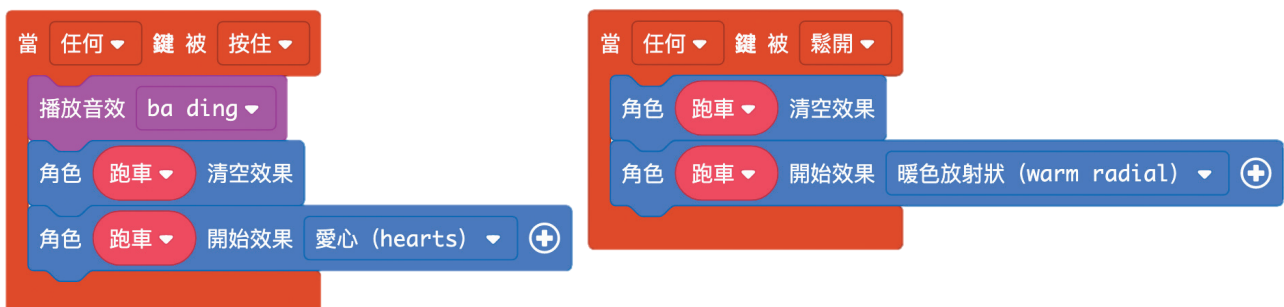
1. 將角色保留在畫面中，不可移動到視窗外



2. 當移動時有音效，並會變換另一種的角色效果

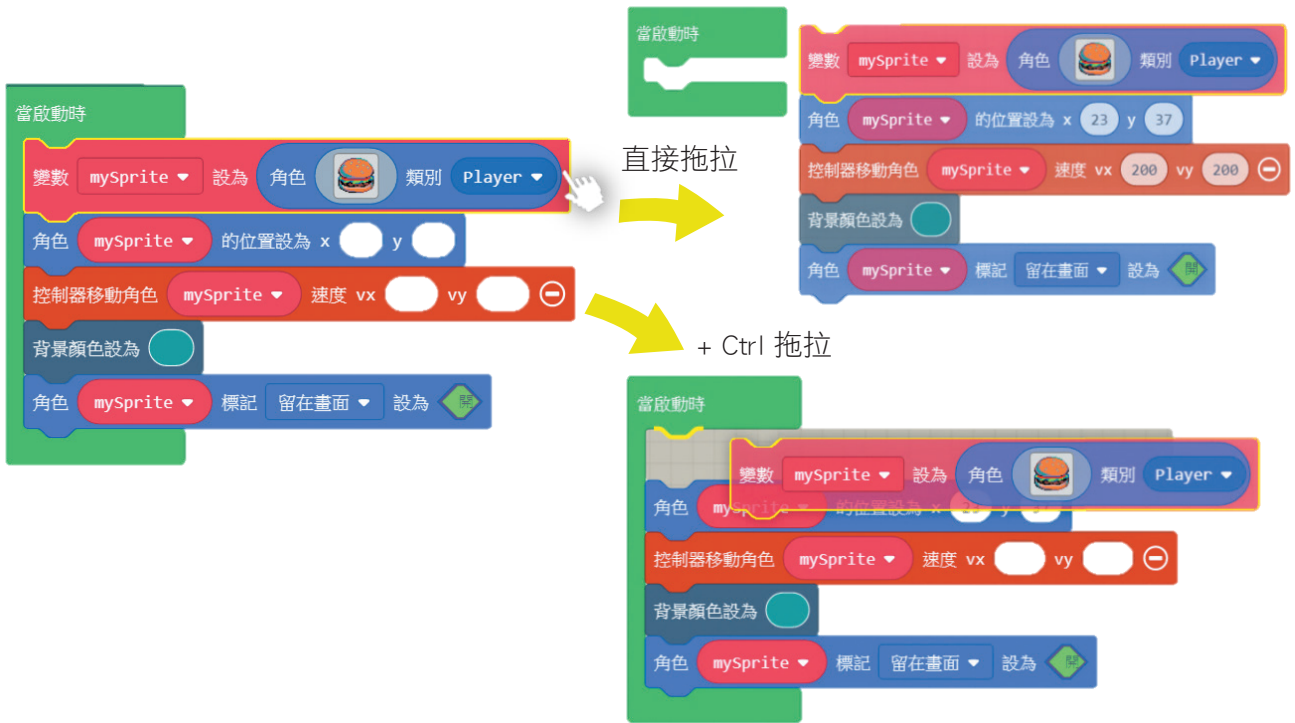


3. 當移動時會出現另一種角色效果，如果停止後又恢復成原來的角色效果



# 1.8 積木編輯技巧


## 一、移動積木



## 二、複製積木



## 1.9 儲存專案和分享

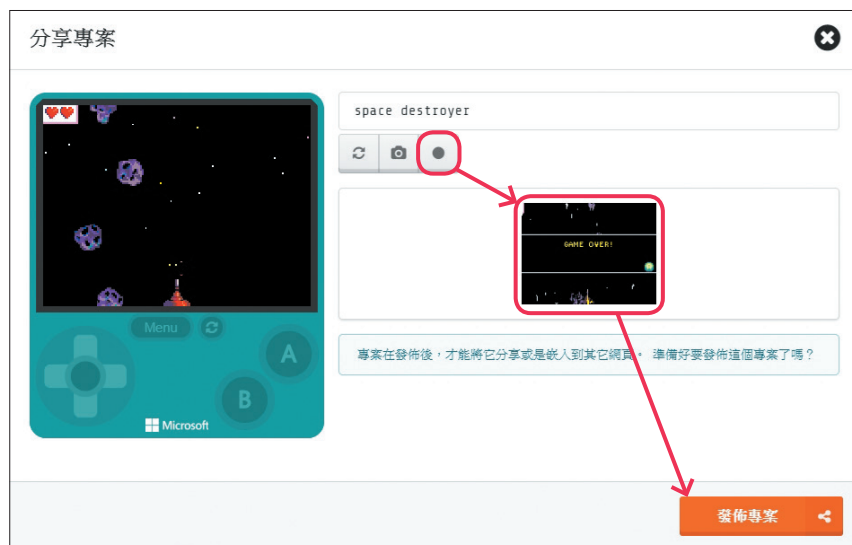
當專案完成後要儲存檔案作為之後還能編輯的檔案時，可以使用程式編輯區下方的存檔鈕 ，儲存成 PNG 檔，或是使用分享的方式，將自己設計的遊戲分享給朋友玩，可使用以下操作方式分享遊戲檔案：

**step 1** 點選模擬器上方的「分享」圖示。



 分享遊戲

**step 2** 出現「分享專案」視窗，可先錄製一段遊戲的 GIF 動畫短片（可省略），再按下「發佈專案」鈕。



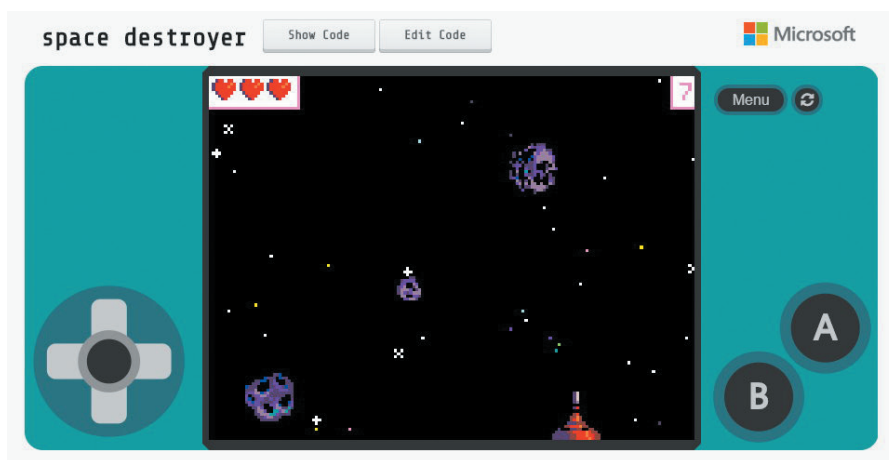
 錄製 GIF 動畫短片

**step 3** 即可將網址或 QR Code 分享給朋友。



🕒 分享專案網址

**step 4** 當收到網址點開後就可以在線上玩你設計的遊戲，或按「Show Code」顯示積木程式，也可以按「Edit Code」修改遊戲積木程式。




🕒 在線上玩設計的遊戲

### Arcade 離線版積木軟體

由於 Microsoft MakeCode 需在連接網際網路時使用，如果沒有網路或網路不順暢時將無法正常使用，目前網路上已有 Arcade 離線版積木軟體，如有需要請自行上網搜尋「MakeCodeArcade 離線版」下載使用。

## 1.10 重置或刪除專案

當專案越來越多時，要清除在「主頁」的「我的所有專案」中所有專案程式時，點選「主頁」右上角  圖示，在下拉清單中點選「重置」即可清除所有專案。



### 重置所有專案

如果只是要刪除某幾個專案程式，可從「主頁」點選「我的所有專案」，進入「我的所有專案」視窗後，勾選要刪除的專案右上角圓圈，再按「刪除」鈕即可刪除選取的專案。



### 刪除專案

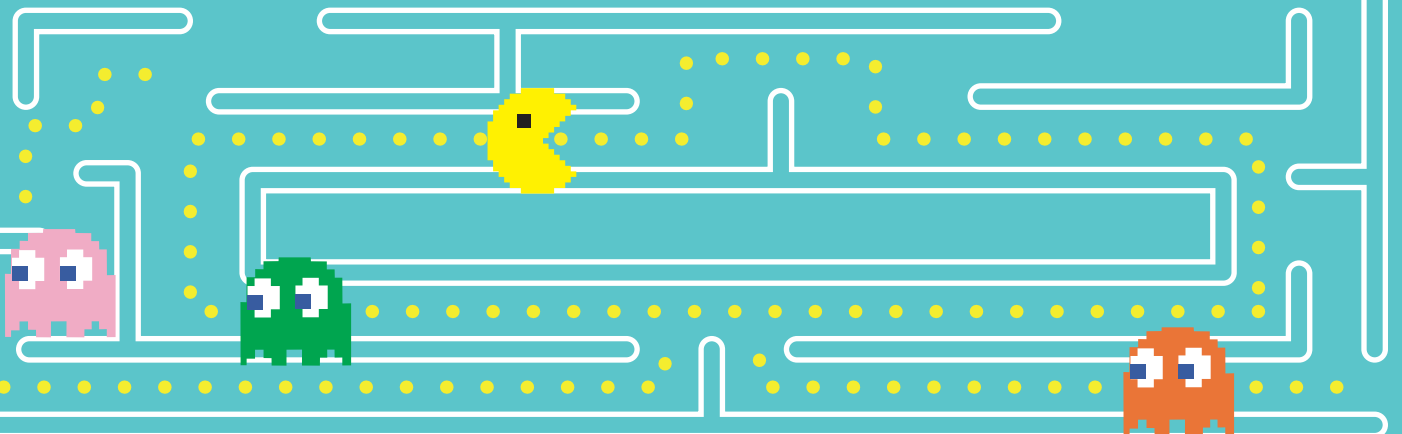
## 練習一下

開啟 MakeCode Arcade 其他範例遊戲，找出一個好玩的遊戲，並使用分享專案的功能，將遊戲分享給你的朋友（請依範例遊戲自行練習分享程式）。

# 2 Chapter

## Arcade 遊戲設計概念

本章利用一個小迷宮遊戲，針對 Arcade 遊戲設計的概念做講解，包括屏幕提示訊息、製作背景圖、設定障礙物（牆）、創建遊戲角色與角色移動、偵測角色類別重疊、遊戲聲音和畫面特效、結束遊戲方法等，將學習到一個完整遊戲所需的應用，後面部分應用對初學者會有一定的難度可先省略，等到有基礎後再回頭來實作練習。



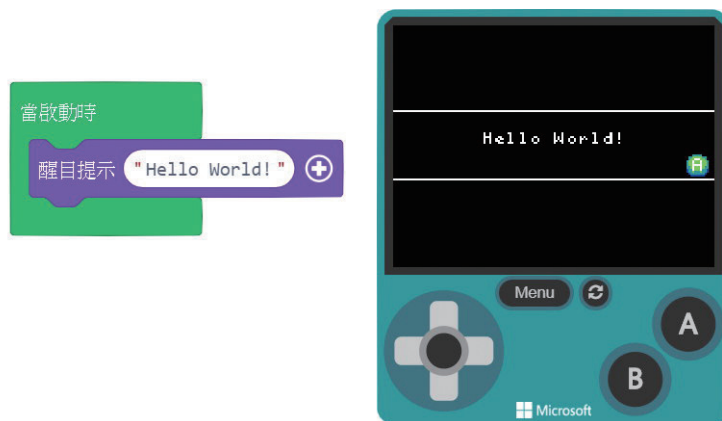
## 遊戲說明

遊戲主角叫小明，在迷宮中尋找食物，當獲得一個食物時將得到一分，遊戲時間設定為 20 秒，如果在時間內獲得 10 分以上，將獲得遊戲的勝利。

## 2.1 事件積木

在執行 Arcade 遊戲時，所有程式積木都必須放在事件積木中，當該事件被觸發時，事件內部的積木程式才會被執行。其中在「迴圈」積木盒中的「當啟動時」事件積木是遊戲機開啟電源或是重置（RESET）後，首先被觸發的事件，且「當啟動時」事件在遊戲運行中只會被執行一次，一般用來做遊戲的初始設定，例如：創建遊戲中的主角、畫面背景或設定變數初始值等。

下圖為從「遊戲」積木盒中拖曳出「醒目提示文字」積木到「當啟動時」事件積木內，並手動輸入「Hello World!」文字後，遊戲開始執行時會在屏幕顯示「Hello World!」文字，並等待按「A」鍵繼續。



● 遊戲開始前顯示醒目提示文字

其他常見的 Arcade 遊戲事件如：「當 A 鍵被按住」、「當角色類別重疊角色類別時」、「當遊戲更新時」、「當遊戲間隔  $\times \times \times$  毫秒時」、「當角色碰到牆壁時」、「當倒數結束時」、「重複無限次」等事件積木，後面章節將一一介紹。

## 2.2 遊戲背景

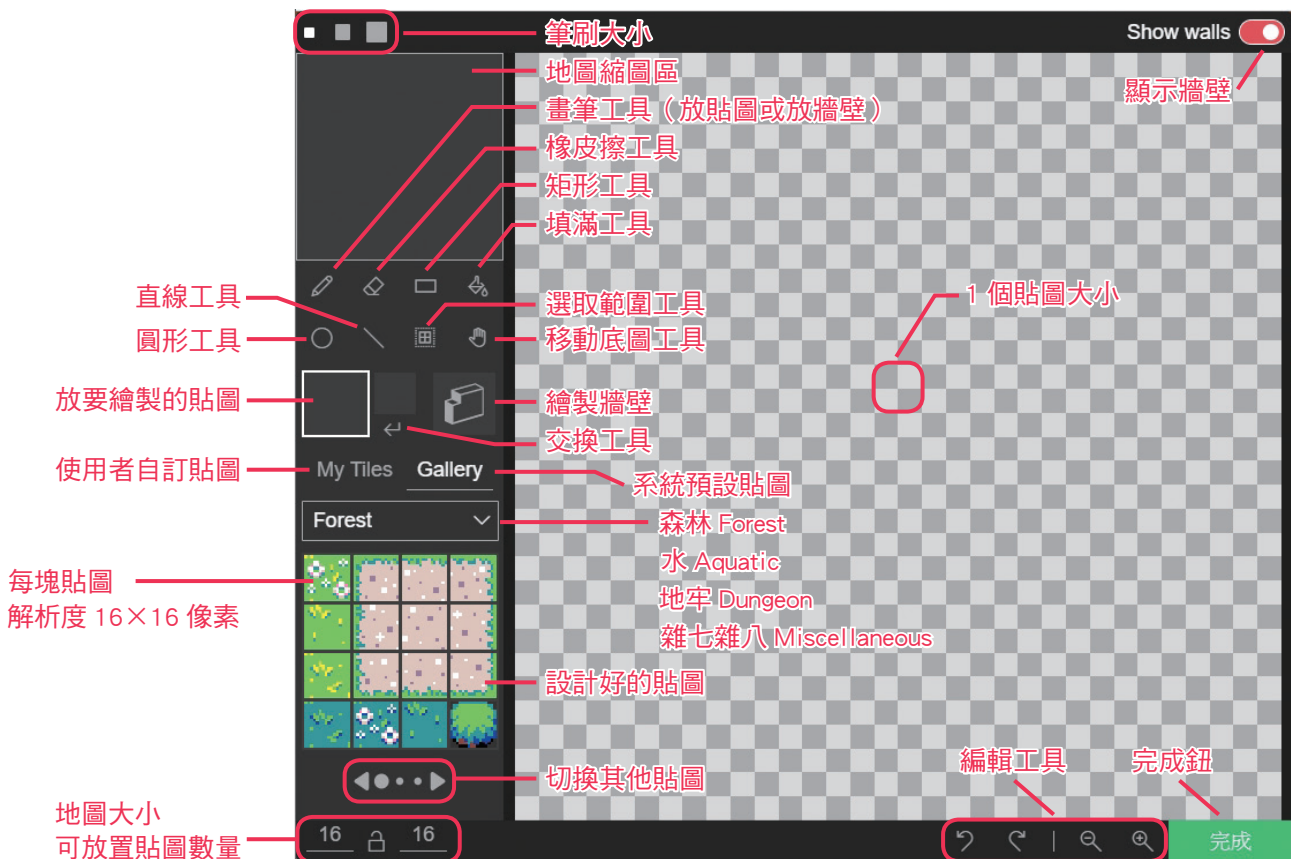
每個好玩的遊戲都會設計一個漂亮的背景和一些吸引人的角色出現在遊戲中，在 Arcade 遊戲的「背景」是指一個不能隨意移動的圖片，可能是填滿顏色的畫面（Screen）或是自行繪製的圖像，也可以是使用預設貼圖所拼接成的貼圖地圖（TileMap）。

在繪製背景前，我們需要先定義遊戲的範圍，如果遊戲範圍超過屏幕大小（160×120 像素），則角色移動時有可能會消失在屏幕外。下面我們使用 Arcade 設計好的貼圖來製作和屏幕差不多大小的貼圖地圖（遊戲範圍）：

**step 1** 從「場景」積木盒中拖曳出「貼圖地圖設為 ...」積木指令，放在「當啟動時」事件中。

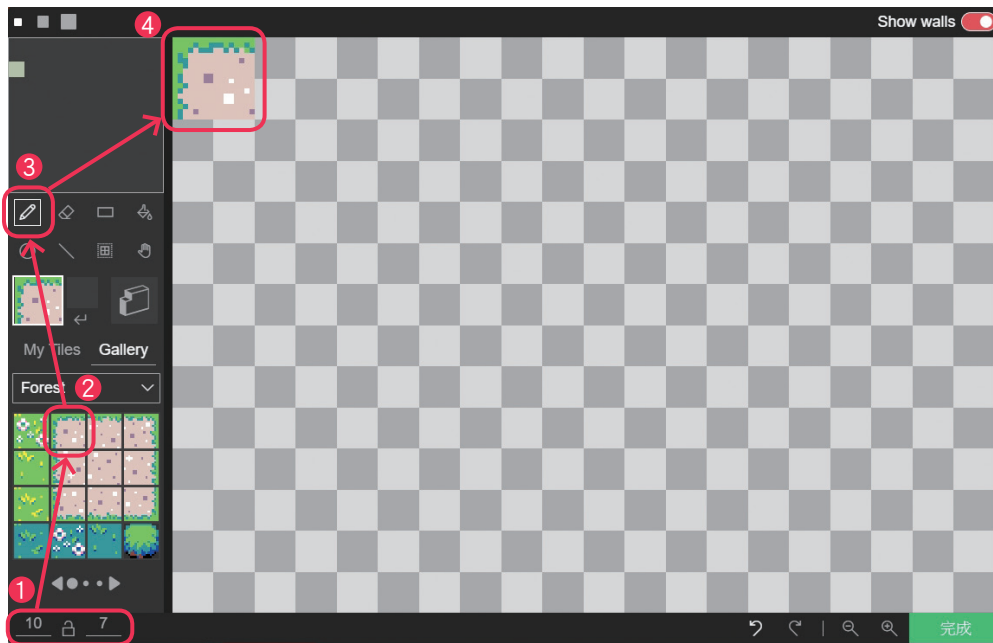


**step 2** 點選積木上的地圖「縮圖」，開啟「地圖編輯器」視窗，右側的「地圖編輯區」大小預設可放置 16×16 塊貼圖（左下角可更改大小），每塊貼圖解析度為 16×16 像素，編輯區中灰色和白色的小方格代表是透明色，一個貼圖佔四個小方格。

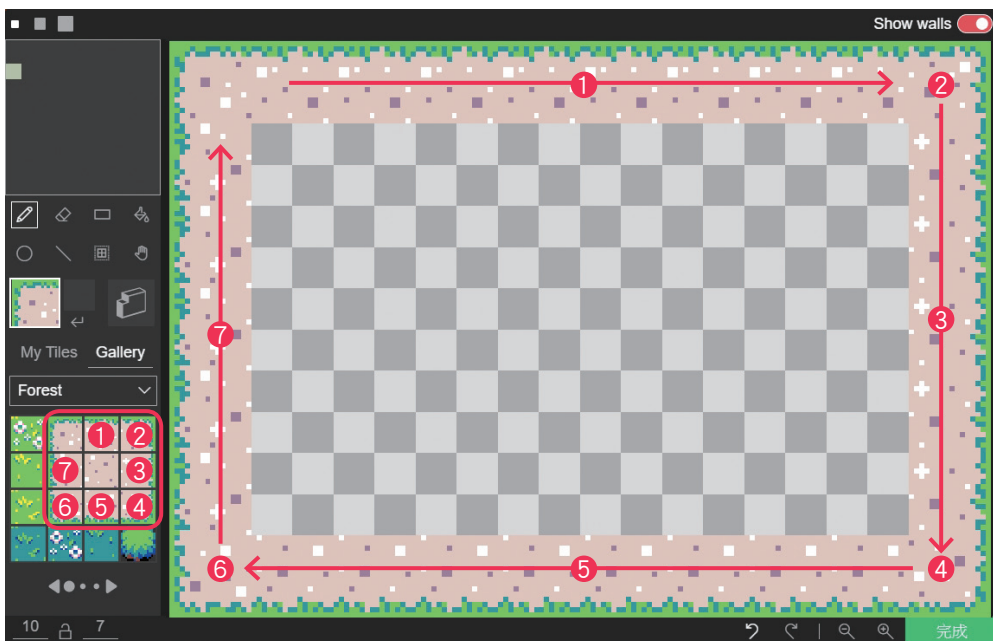


**step 3** 使用 Arcade 貼圖製作背景圖時的每塊貼圖大小是  $16 \times 16$  像素，為了不超過屏幕，所以，我們將製作的背景設定為「 $10 \times 7$ 」塊貼圖來組合出  $160 \times 112$  像素的背景圖後，請依以下順序操作：

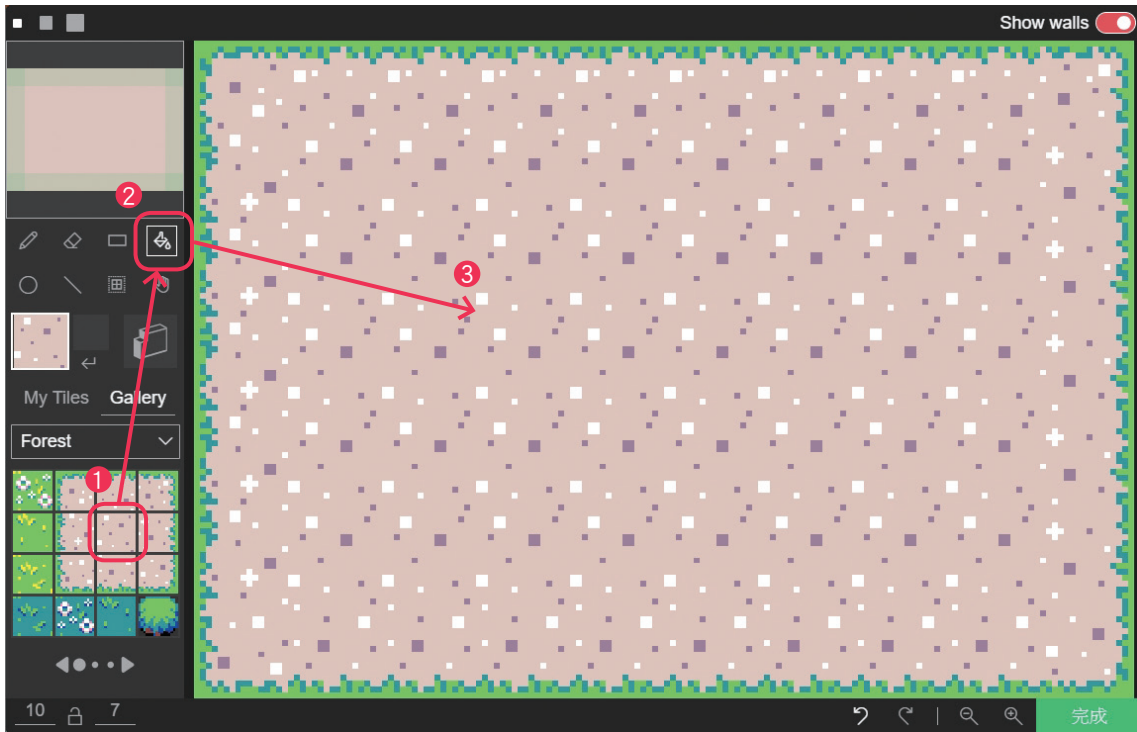
- 1 設定地圖貼圖大小為「 $10 \times 7$ 」塊貼圖
- 2 選取系統預設圖庫「Forest」（森林）的左上角草地貼圖
- 3 選擇「畫筆工具」
- 4 將左上角草地貼圖繪製在「地圖編輯區」的左上角



**step 4** 依序將下圖編號 1~7 的草地貼圖繪製於「地圖編輯區」的四周。



**step 5** 選取「地面」貼圖後，填滿「地圖編輯區」的中間區域。



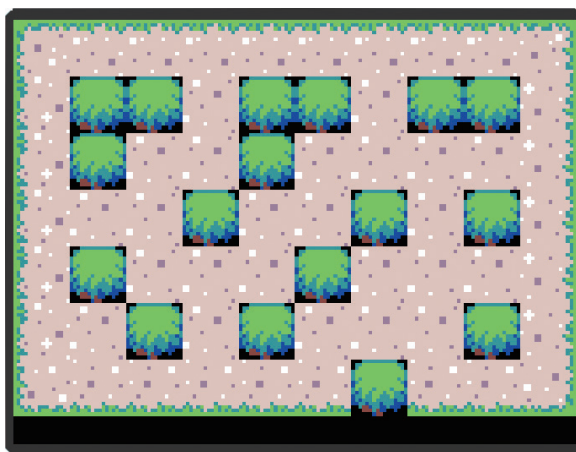
**step 6** 選取「森林」貼圖後，畫出任意的迷宮，因為是小型迷宮「森林」盡量不要太密集。



**step 7** 為避免角色穿過「森林」貼圖，選取「牆」貼圖後，將所有的「森林」貼圖標示為「牆」，如要取消牆的標示，需在選取「牆」貼圖後，使用「橡皮擦」工具清除標示。

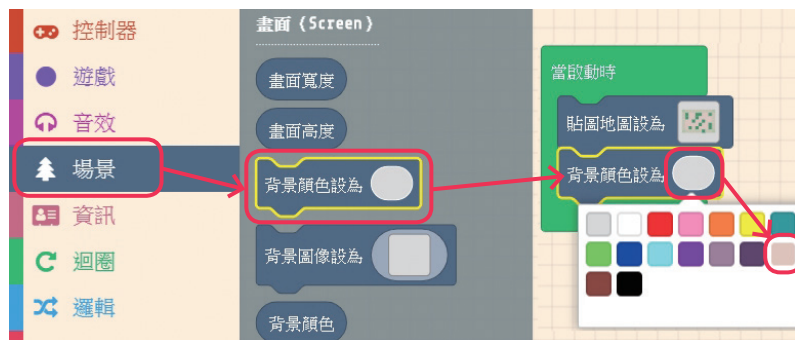


**step 8** 點選「完成」鈕回到程式編輯畫面，此時屏幕出現剛剛繪製的地圖。

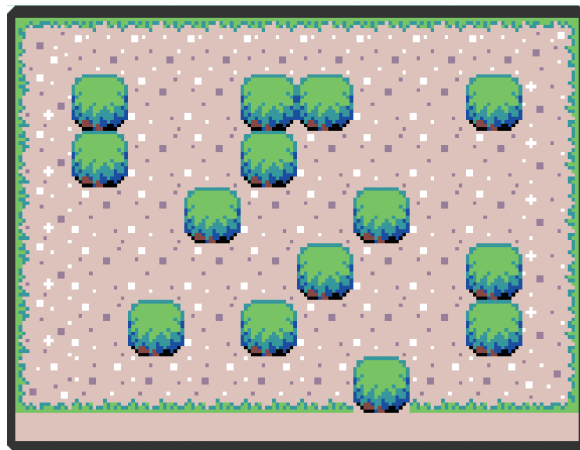


**step 9** 上面背景圖的「森林」貼圖邊緣有黑邊，是因為「森林」貼圖的邊緣是透明色（屏幕背景色預設為黑色），因此，我們給背景色填滿一個與貼圖顏色相近的顏色，即可避免貼圖有黑邊的現象。

- ① 從「場景」積木盒中拖曳出「背景顏色設為 ...」積木指令，放在「當啟動時」事件中。
- ② 點選背景顏色圖示，並選擇「#e5cdc4」的顏色方塊。



- ③ 完成的背景圖「森林」貼圖邊緣就沒有黑邊了



## 2.3 遊戲角色與角色移動

所謂角色 (Sprites) 是指遊戲中可獨立控制、移動的圖像，可能是遊戲的主角、敵人、食物或是一顆子彈，所有的角色必須給予一個名字（不能重複）和類別，方便對其做控制和碰撞偵測，並且角色是以圖像呈現在屏幕上，以下為創建角色的步驟：

**step 1** 拖曳「角色」積木盒中的「變數設為角色」積木指令至「當啟動時」事件積木中。



**step 2** 點選「變數設為角色」積木的角色「縮圖」，開啟「繪圖編輯器」視窗後，切換至「素材庫」，並點選一個最小的素材圖（因為地圖範圍較小，請勿選大的素材圖）。



**step 3** 下拉變數名稱清單選項，選擇「重新命名變數」更改角色名稱為「小明」。

「新的變數」選項：只改變目前積木上的變數名稱。

「重新命名變數」選項：改變程式中所有已存在相同的變數名稱。



**step 4** 為了可以控制角色移動，我們需要將角色跟控制器做連結。

- ① 拖曳「控制器」積木盒中的「控制器移動角色」積木指令至「當啟動時」事件積木中。
- ② 下拉角色名稱清單，選擇「小明」。



**step 5** 請試著使用模擬器左側的方向鍵移動角色，如果要使用鍵盤的上、下、左、右鍵移動角色，滑鼠游標要滑過模擬器，使模擬器上的控制鈕呈現高亮度才能使用鍵盤控制角色。



**step 6** 我們準備在屏幕左下角新增食物，如果小明去取得食物，則得分加一。

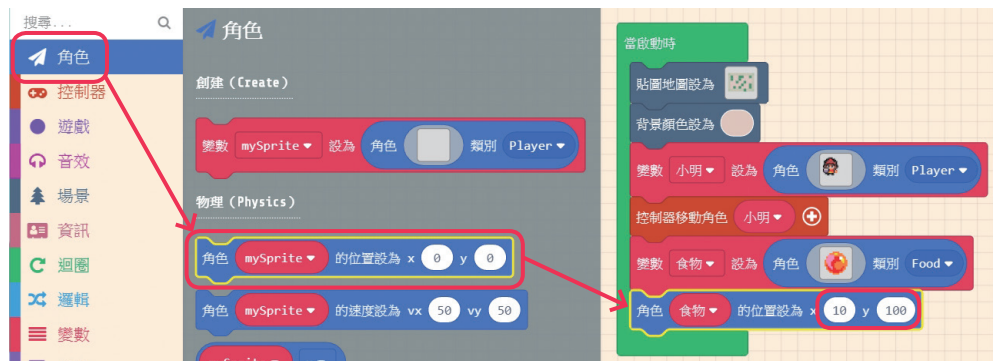
首先創建一個名為「食物」的角色。

- ① 拖曳「角色」積木盒中的「變數設為角色」積木指令至「當啟動時」事件積木中。
- ② 點選「變數設為角色」積木的角色「縮圖」。
- ③ 開啟「繪圖編輯器」視窗後切換至「素材庫」。
- ④ 到素材庫的最後面，點選「珠子」貼圖。
- ⑤ 將角色重新命名為「食物」，類別設為「Food」。



**step 7** 將角色「食物」放置在屏幕的左下角。

- ① 拖曳「角色」積木盒中的「角色位置設為 ...」積木至「當啟動時」事件積木中。
- ② 下拉角色名稱清單，選擇「食物」。
- ③ 位置更改為  $x=10$ 、 $y=100$ 。



**step 8** 角色完成後如下圖。



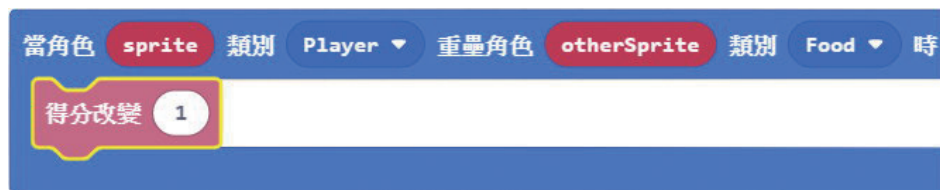
## 2.4 偵測角色類別重疊

Arcade 採用的碰種偵測是某一角色類別重疊到另一角色類別時，會觸發「重疊角色」事件，在「重疊角色」事件中要操作觸發角色的行為必須使用類別前面的角色代稱（Sprite 或 otherSprite）來操作。

**step 1** 拖曳「角色」積木盒中的「當角色類別重疊角色類別時」事件積木至程式編輯區中，並將兩個角色類別設定為當「Player」重疊「Food」時觸發重疊事件。

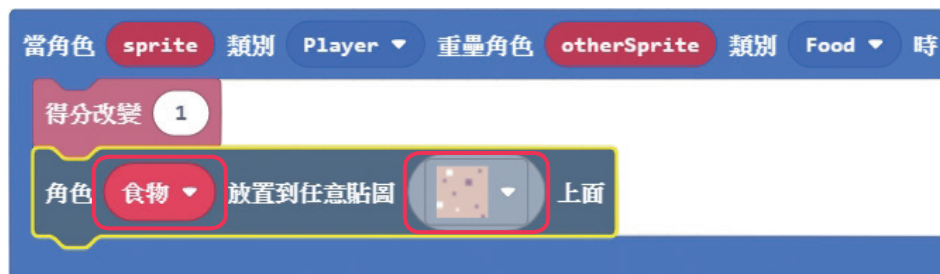


**step 2** 拖曳「資訊」積木盒中的「得分改變」積木至「當角色類別重疊角色類別時」事件積木中。



**step 3** 當小明取得食物時，除了得一分外，我們讓角色「食物」隨機的重新出現在其他「地面」貼圖之處。

- ① 拖曳「場景」積木盒中的「角色放置到任意貼圖上面」積木指令至「當角色類別重疊角色類別時」事件積木中。
- ② 下拉角色名稱清單，選擇「食物」。
- ③ 貼圖選擇「地面」貼圖（必須是在背景圖中存在的貼圖）。



## 2.5 遊戲聲音和畫面特效

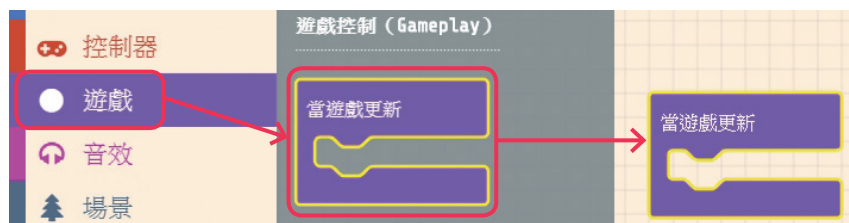
我們讓小明取得食物得分時發出音效，來增加遊戲趣味性。拖曳「音效」積木盒中的「播放音效」積木指令至「當角色類別重疊角色類別時」事件積木中即可。



接下來為較進階的遊戲設計技巧，初學者可先跳過，等有基礎後再回來實作練習。我們將增加整個畫面的特殊視覺效果，雖然可以使用「場景」積木盒中的「效果用於開始畫面」積木指令，將該積木放置於「當啟動時」事件積木中即可有不錯的畫面效果，但下面我們自己來做一個下雪的畫面特效，讓自製的雪花從屏幕上面隨機位置落下：

**step 1** 拖曳「遊戲」積木盒中的「當遊戲更新」事件積木至程式編輯區。

「當遊戲更新」事件積木內的程式會快速的重複一直執行，在這個事件內應避免使用「暫停 xx 毫秒」積木指令，如果需要使用「暫停 xx 毫秒」積木指令，則改為使用「迴圈」積木盒中的「重複無限次」事件積木。

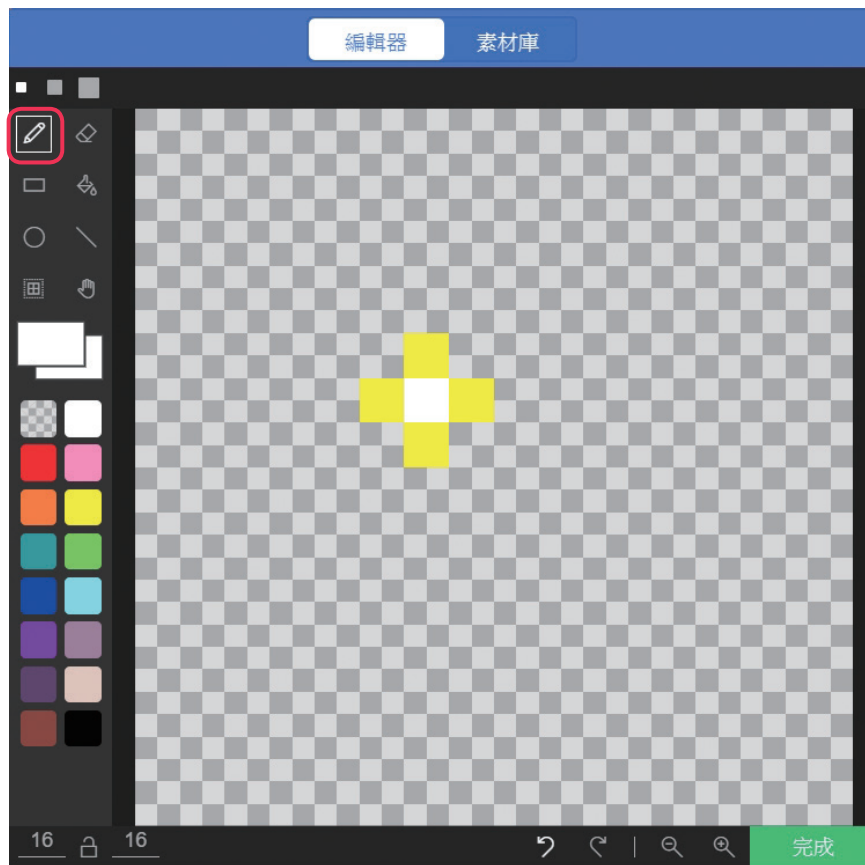


**step 2** 在屏幕上方隨機位置產生雪花落下。

- ① 拖曳「角色」積木盒中的「變數設為發射物來自側邊」積木指令至「當遊戲更新」事件積木中。
- ② 更改變數名稱為「雪花」。
- ③ 輸入發射物的水平速度 ( $v_x=0$ )，垂直速度 ( $v_y=30$ )。



- ④ 點選上圖「雪花」的發射物「縮圖」，開啟「圖形編輯器」，使用「畫筆工具」繪製一朵雪花。



- ⑤ 拖曳「角色」積木盒中的「角色位置設為...」積木指令至「當遊戲更新」事件積木中，下拉角色名稱清單，選擇「雪花」。



- ⑥ 拖曳「數學」積木盒中的「隨機取數」積木指令至角色「雪花」的水平位置 (x)。



- ⑦ 拖曳「場景」積木盒中的獲取「畫面寬度」積木指令至「隨機取數」積木的結束值，如此，雪花將會在屏幕上方任一位置隨機產生。



- ⑧ 此時畫面已有出現「雪花」效果，但雪花會被設定為牆的「森林」貼圖擋住。



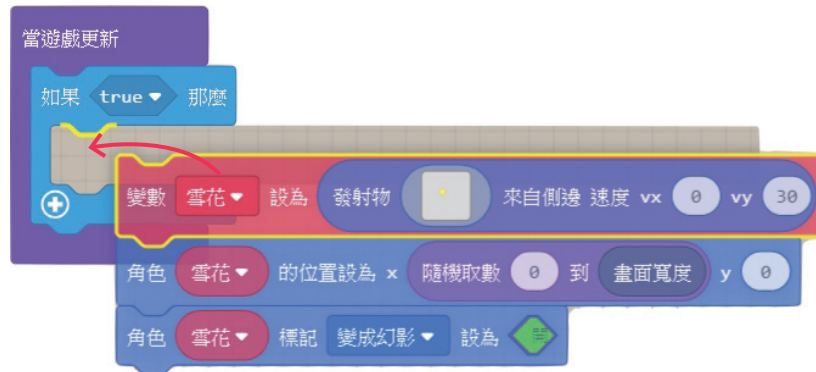
**step 3** 將發射物「雪花」設定為幻影就不會被牆所擋住了。

- ① 拖曳「角色」積木盒中的「角色標記..」積木指令至「當遊戲更新」事件積木中。
- ② 下拉角色名稱清單選擇「雪花」。
- ③ 下拉標記清單選擇「變成幻影」，並將效果設為「開」。



**step 4** 目前的「雪花」有點太多，因為「當遊戲更新」的事件在遊戲執行中會一直快速不斷的重複觸發，每觸發一次便產生一個「雪花」角色從上面往下掉，我們將使用 25% 的機率來決定是否每次觸發「當遊戲更新」的事件時都要產生「雪花」。

- ① 拖曳「邏輯」積木盒中的「如果 ... 那麼」積木指令至「當遊戲更新」事件積木中。
- ② 將原來的積木程式拖曳至「如果 ... 那麼」積木中。



- ③ 拖曳「數學」積木盒中的「隨機取布林值 true 的機率設為 ...」積木至「如果」積木的條件判斷中，並將機率設為「25%」。



- ④ 完成後的雪花就只有原來的 1/4 了。



## 2.6 更換背景貼圖

有時候我們希望當得分超過某個分數時能自動更換場景，使用更換背景貼圖可達到相同的效果，做法是讓程式將要更換的背景貼圖列舉後一一更換成指定的貼圖。為簡化程式，以下操作步驟是當按 A 鍵時，將背景圖中的「地面」貼圖更換成「草地」貼圖：

**step 1** 拖曳「控制器」積木盒中的「當 A 鍵被按住」事件積木至程式編輯區。



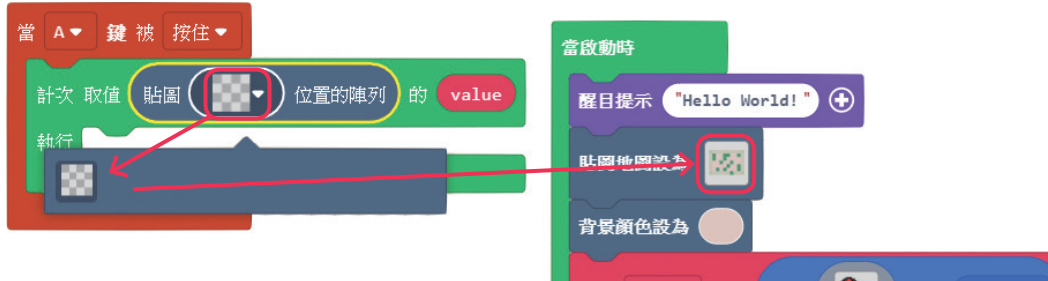
**step 2** 拖曳「迴圈」積木盒中的「計次取值」積木指令至「當 A 鍵被按住」事件積木中。



**step 3** 拖曳「場景」積木盒中的獲取「貼圖位置的陣列」積木指令至「計次取值」積木取代「list」參數，做為取值的來源參數，當程式執行到「計次取值」積木時，背景圖上所有與指定貼圖相同的貼圖，其座標將一一的放入「value」變數中。



**step 4** 下拉貼圖清單後，如果貼圖清單中沒有出現所需的貼圖，則先去「貼圖地圖設為...」積木指令，點選地圖「縮圖」，開啟「地圖編輯器」，並在「地圖編輯器」中點一下需要用到的貼圖，瀏覽器就會記錄住該貼圖。



**step 5** 再次回到「貼圖位置的陣列」積木貼圖下拉清單，就會看到剛才曾經點過的貼圖，再點選「地面」貼圖做為要被取代的貼圖。



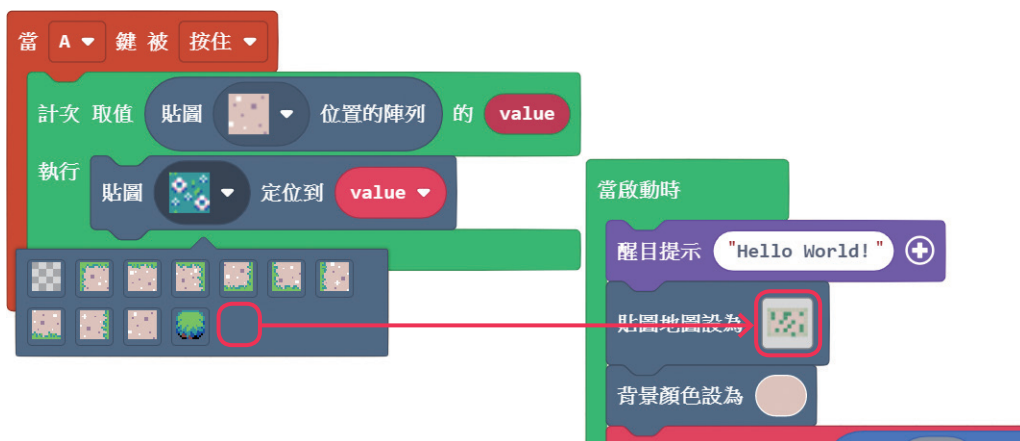
**step 6** 要取代的貼圖則使用「場景」積木盒中的「貼圖定位到...」積木指令，先放至「計次取值」積木內。



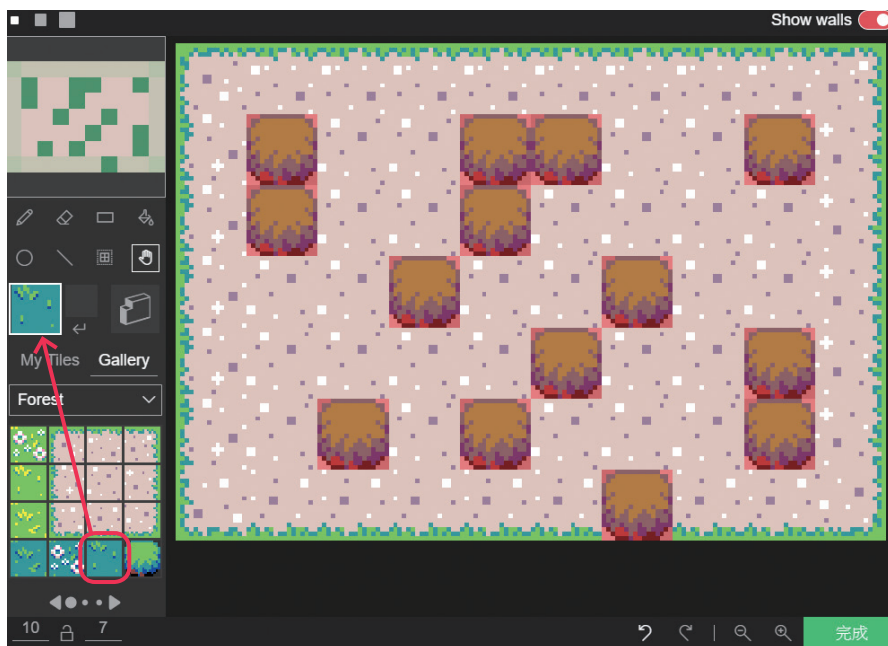
**step 7** 再將「計次取值」積木的「Value」變數拖曳到「貼圖定位到...」積木的位置參數。



**step 8** 下拉貼圖清單，如果沒有我們需要的「草地」貼圖，則先去「貼圖地圖設為 ...」積木指令，點選地圖「縮圖」，開啟「地圖編輯器」。



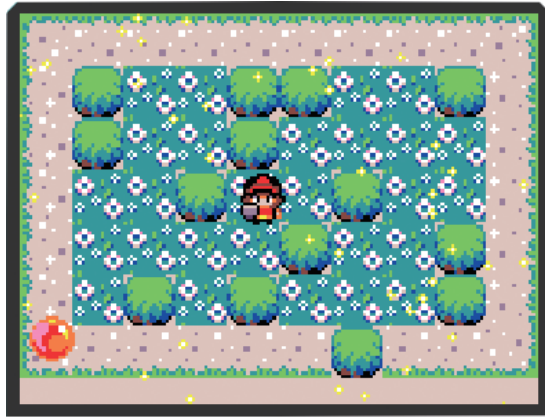
**step 9** 在「地圖編輯器」中點一次需要用到的「草地」貼圖，瀏覽器會記錄住該貼圖。



**step 10** 再次點「貼圖定位到 ...」積木下拉貼圖清單，就會看到剛才的貼圖，點選「草地」貼圖做為要成為取代的貼圖。



**step 11** 完成背景貼圖取代，在遊戲過程中只要按下「A」鍵，「地面」貼圖立即被取代成為「草地」貼圖。



## 2.7 結束遊戲

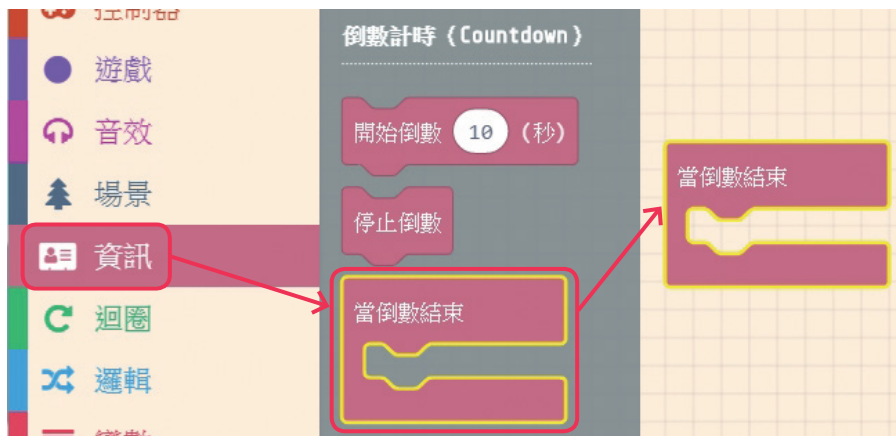
由於目前的程式一旦執行遊戲就不會結束，因此，我們設定遊戲時間為 20 秒，看看在 20 秒內小明能得到幾分。

拖曳「資訊」積木盒中的「開始倒數」積木至「當啟動時」事件積木中，將「開始倒數」積木的秒數改為「20」秒。

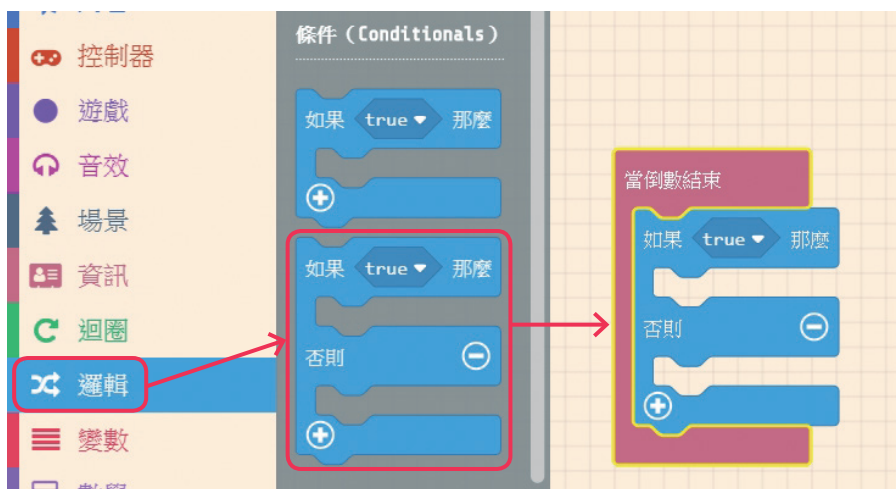


為了增加遊戲的樂趣，在「20」秒遊戲結束後如果小明得到超過 10 分，則顯示「YOU WIN」否則顯示「GAME OVER」。

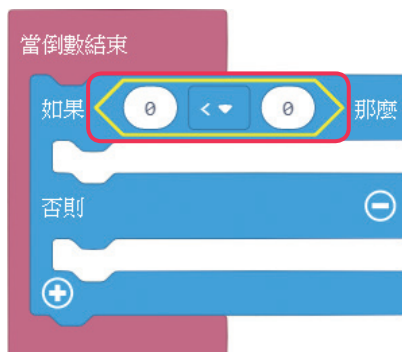
**step 1** 拖曳「遊戲」積木盒中的「當倒數結束」事件積木到「程式編輯區」中。



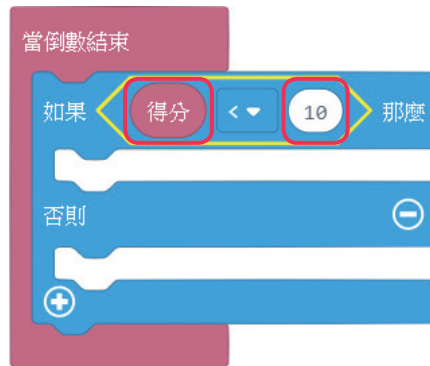
**step 2** 拖曳「邏輯」積木盒中的「如果 ... 否則 ...」積木到「當倒數結束」事件積木中。



**step 3** 拖曳「邏輯」積木盒中的「小於」積木到「如果 ... 否則 ...」積木的條件中。



**step 4** 拖曳「資訊」積木盒中的「得分」積木到「小於」積木的左邊參數中，並將「小於」積木的右邊參數改為「10」。



**step 5** 拖曳「遊戲」積木盒中的「遊戲結束」積木到「如果...否則...」積木的「條件成立」程式區中。



**step 6** 在「遊戲結束」積木上按右鍵，選擇「複製」，並將複製出來的「遊戲結束」積木拖曳到「如果...否則...」積木的「條件不成立」程式區中。



**step 7** 將在「條件不成立」程式區中的「遊戲結束」積木改成「贏」，並增加撒「紙片」效果。



## 2.8 完成的積木程式碼

〔程式檔：02 小迷宮〕



程式碼續接下頁



## 2.9 在程式編輯區中常用的按鍵

功 能	按 鍵
放大積木	Ctrl + 滑鼠滾輪往前
縮小積木	Ctrl + 滑鼠滾輪往後
刪除積木	點選積木後按 Delete 或是拖曳至「積木指令區」
拖曳單塊積木	Ctrl + 滑鼠左鍵 Alt + 滑鼠左鍵
復原上一步	Ctrl + z
複製積木	Ctrl + c
剪下積木	Ctrl + x
貼上積木	Ctrl + v

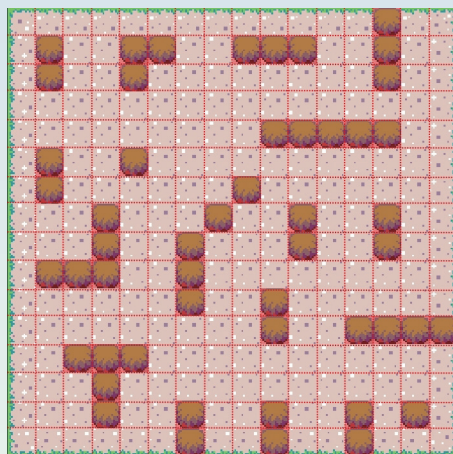
### 練習一下

讓畫面特效中的「雪花」飄下來是從右上往左下掉下來。

# 綜合實作題

**題目名稱** 迷宮尋寶遊戲

**題目說明** 請參考「附錄二、範例 18」，將本章「小迷宮」更改為 16×16 貼圖地圖迷宮（如下圖），遊戲倒數計時增加為 30 秒，並在迷宮中增加另外二種食物。



## 解題提示

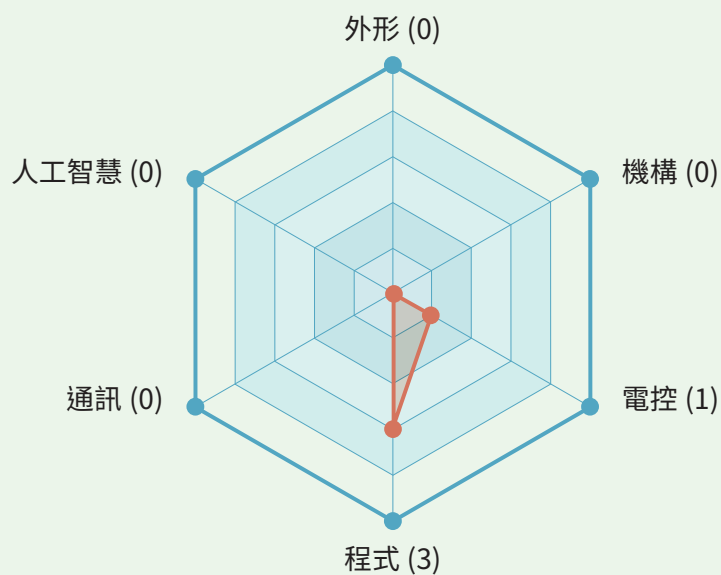
在「當啟動時」事件中，新增「小明」角色後，增加「場景／角色跟隨」積木指令，並再複製出兩組「食物」角色積木指令。

創客題目編號：A033001

## 創客指標

外形	0
機構	0
電控	1
程式	3
通訊	0
人工智慧	0
<b>創客總數</b>	<b>4</b>

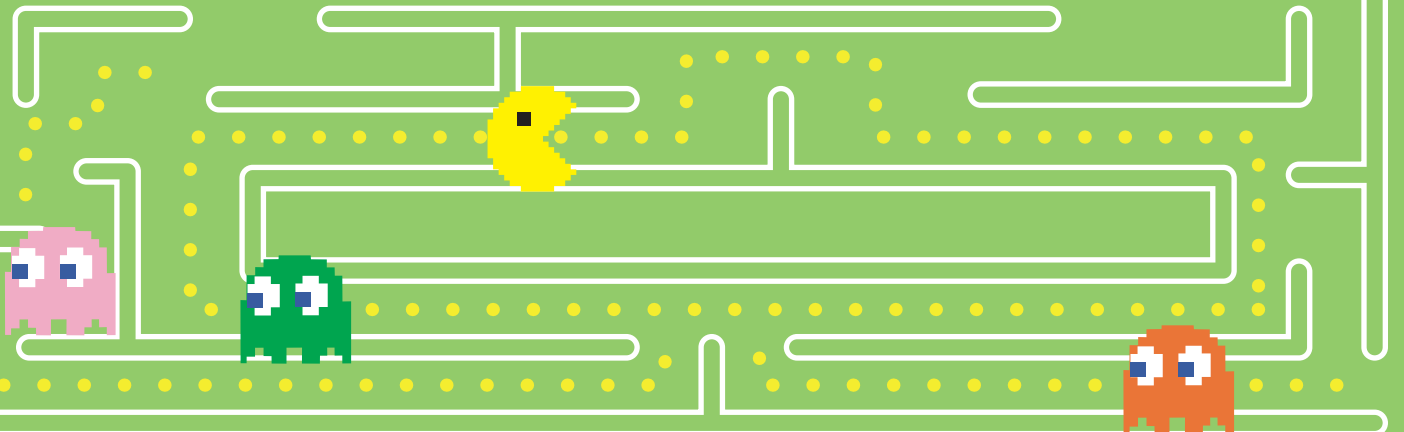
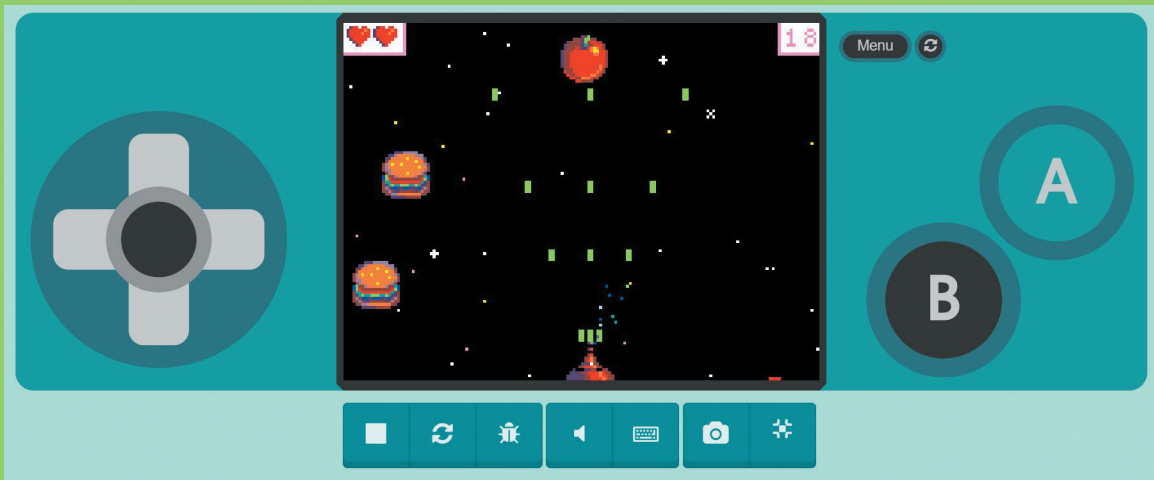
45  
mins





## 修改「太空驅逐艦」 (Space Destroyer) 範例

本章從 MakeCode Arcade「主頁」中的「程式做的積木遊戲」開啟「太空驅逐艦」(Space Destroyer) 範例，來修改遊戲，學習更詳細的 Arcade 遊戲設計，避免初學者覺得遊戲設計難以入門，事實上使用 MakeCode Arcade 從無到有的遊戲設計其實很簡單。

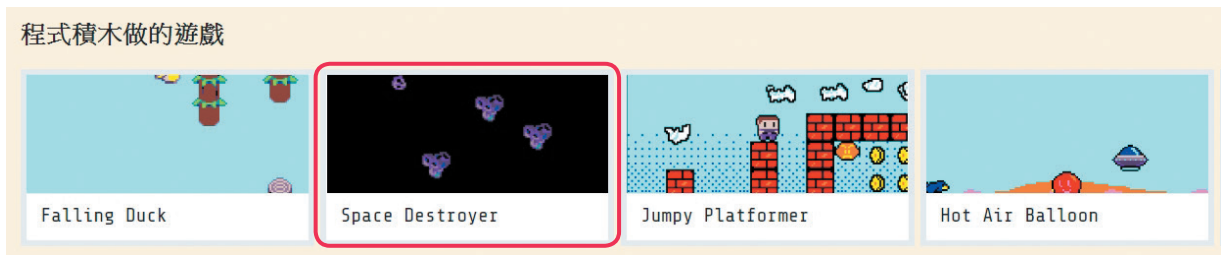


## 遊戲說明

「太空驅逐艦」是一款射擊遊戲，遊戲主角是一艘太空船，太空每隔 0.5 秒會落下一顆隨機大小的隕石，玩家需要使用遊戲機的方向鍵控制太空船閃躲隕石或是按「A」鍵發射砲彈來擊毀隕石，當太空船發射的砲彈射中 1 顆隕石可獲得 1 分；當太空船被隕石擊中時生命值將減 1（開始遊戲生命值設為 3），生命值為 0 時遊戲結束。

## 3.1 開啟積木做的遊戲範例程式

在 MakeCode Arcade 編輯器「主頁」點選「程式積木做的遊戲」區塊中的太空驅逐艦積木遊戲「Space Destroyer」。



「太空驅逐艦」積木遊戲

畫面顯示遊戲 GIF 動畫短片，點選「打開範例」。



打開「太空驅逐艦」範例

## 3.2 事件積木

所有要執行的積木程式都要放在事件類型的積木中，當某個事件被觸發時才會去執行該事件內的積木程式，其中「當啟動時」的事件積木是遊戲機開啟電源或是重置（RESET）後，首先觸發的事件，且「當啟動時」事件在遊戲運行中只會被執行一次，一般用來做遊戲的初始設定，例如：創建遊戲中的主角、畫面背景或設定變數初始值。

依遊戲規則，這個遊戲有使用到五個事件：



### 3.3 「當啟動時」事件積木 (遊戲初始設定)

依遊戲說明「太空驅逐艦」遊戲一開始，做以下的初始設定：

#### 程式碼說明

##### ▼行號 01

創建遊戲中會使用到的「隕石」圖片物件。

##### ▼行號 02

創建遊戲主角一艘「太空船」，類別為「Player」。

##### ▼行號 03

設定「太空船」只能在屏幕內移動。

##### ▼行號 04

設定「太空船」的初始位置在屏幕下方。

##### ▼行號 05

設定遊戲機控制器與「太空船」做連結，以便能使用方向鍵控制太空船移動。

##### ▼行號 06

設定「太空船」的生命值為 3。

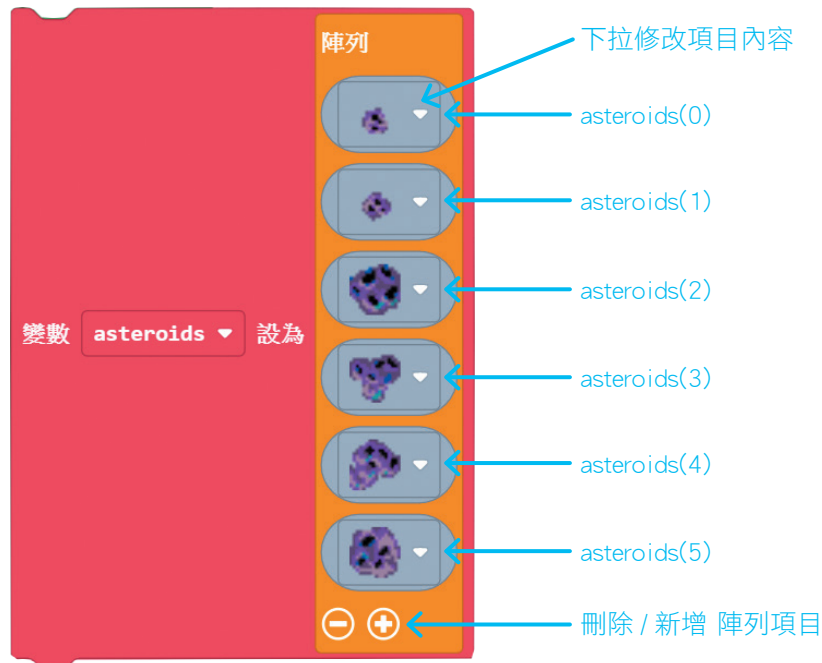
##### ▼行號 07

給遊戲一個很炫的背景。

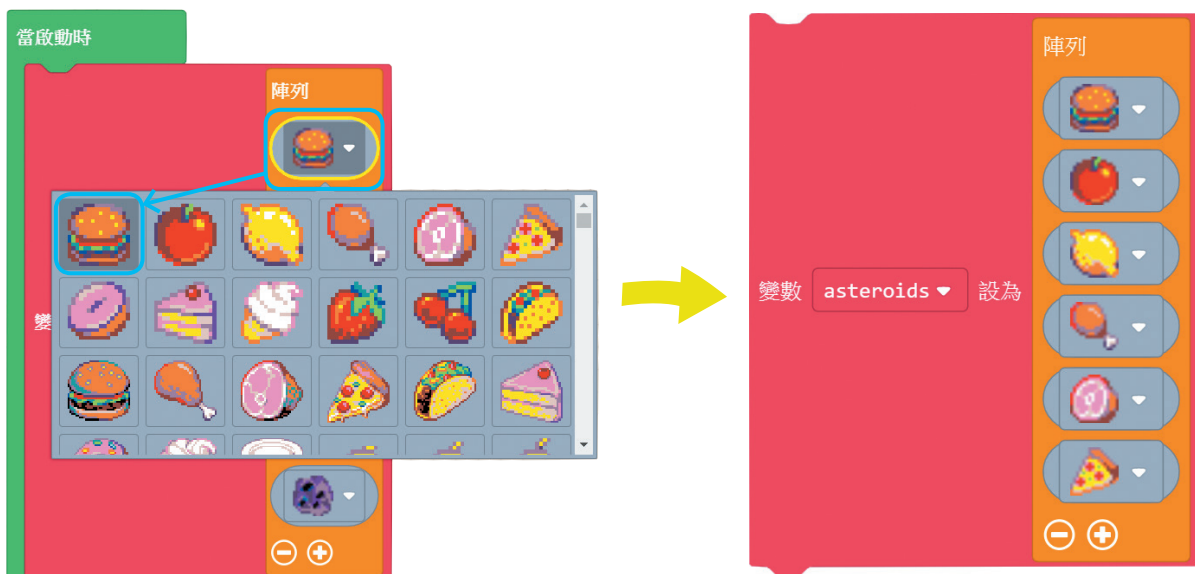


## 1 積木一

在「太空驅逐艦」範例的「當啟動時」事件中，第一個積木指令為「進階 / 陣列」積木盒中的「變數設為陣列」積木指令，並將陣列變數命名為「asteroids」（小行星），陣列主要的用途是建立一份相同名稱的項目（元素）清單，當設定好清單中的項目內容後，在遊戲執行中透過「索引值」可以很方便的從清單中取得指定的項目內容，這裡的這份「asteroids」陣列清單是用來存放遊戲會用到的隕石圖片物件，分別為 asteroids(0) ~ asteroids(5) 的 6 個不同隕石圖的項目內容。



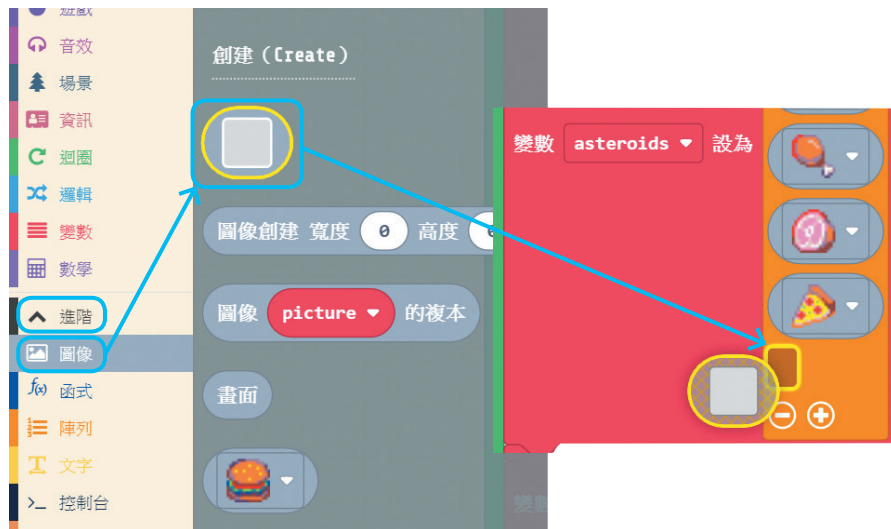
**step 1** 點選隕石圖後可從 Arcade 素材庫中選擇其他的圖檔替換目前的隕石圖，請試著將六個隕石圖案更改為六個食物圖案。



**step 2** 按陣列積木下方的「+」會在清單中增加一個項目，新增的項目內容預設為無內容 (NULL)，可以從「積木區」的「進階 / 圖像」積木盒中拖曳出「一個圖像」到新增的項目內。



**step 3** 如果想要自己繪製圖案，也可從「積木區」的「進階 / 圖像」積木盒中拖曳出「一個空白圖像」到新增的項目清單內。



**step 4** 點選剛才新增的空白圖像後將自動開啟「圖形編輯器」，在圖形編輯器內可以繪製自己想要的圖形，編輯區中白色和灰色組成的方格代表透明色，四個小方格實際上為屏幕上的一個點（像素），Arcade 屏幕解析度預設為  $160 \times 120$  像素，自定義圖形時需注意圖形解析度，避免圖形太大或太小，圖形編輯器左下方可設定圖形解析度（單位：像素）。

按陣列積木下方的「-」會刪除清單中的最後一個項目，請將剛剛新增的項目刪除。



## 2 積木二

第二個積木用來創建遊戲主角，使用「角色」積木盒中的「變數設為角色」積木指令，這個積木指令會在遊戲中創建一個新的角色，並給定這個角色名稱叫「ship」，角色類別預設為「Player」，角色是存在於屏幕上可移動的個體圖像。為了方便理解程式，在「ship」變數名稱上按滑鼠左鍵，選擇「重新命名變數」，並改成「太空船」，積木的角色類別預設為「Player」，也可以自訂角色類別，Arcade 的角色類別主要作為碰撞偵測使用。



### 3 積木三

第三個積木是設定角色在屏幕中移動時所呈現的物理狀態，當設為「開」時，會依選擇的標記方式呈現，可使用多個積木為同一個角色同時設定不同的物理狀態。



#### 設定屬性

留在畫面	設定角色只能在屏幕中移動。
變成幻影	設定角色為虛擬物件不會和其他物件產生碰撞。
自動銷毀	設定當角色移出屏幕外時角色自動刪除。
撞牆時銷毀	當角色碰到設定為牆的圖形或碰到貼圖地圖的邊緣時，角色自動刪除。
撞牆時反彈	當角色自主移動，碰到設定為牆的圖形或碰到貼圖地圖的邊緣時反彈，反彈的角度為 90 度。
顯示座標與速度	在角色下方顯示座標 (x,y) 與速度 (vx,vy)。
不可見	設定角色在屏幕中為隱藏狀態，與幻影相似只是看不到角色。
相對於鏡頭	當「場景」設定鏡頭跟隨 A 角色時，可設定 B 角色出現在在 A 角色的相對位置上。

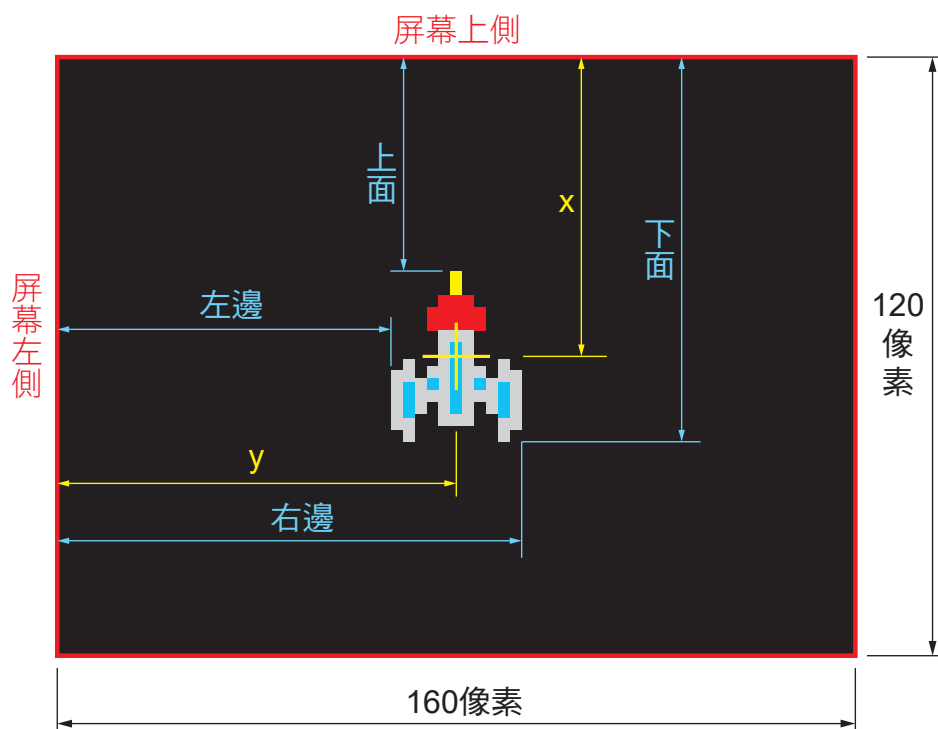
### 4 積木四

第四個積木是「角色」積木盒中的「設定角色屬性」積木指令，創建角色時角色圖片的中心點預設位置是在屏幕的正中央，此「設定角色屬性」積木指令設定角色圖片的下邊緣距離屏幕上方 120 像素（屏幕最底部），這個「設定角色屬性」的積木可設定的屬性包含位置、速度、生命週期與深度等。



## 設定屬性

x	角色的圖形正中心與屏幕左側的像素距離（參考下圖）。
y	角色的圖形正中心與屏幕上方的像素距離（參考下圖）。
速度 vx	設定角色的水平移動速度（正值向右，負值向左）。
速度 vy	設定角色的垂直移動速度（正值向下，負值向上）。
加速度 ax	設定角色的水平漸進移動速度（正值向右，負值向左）。
加速度 ay	設定角色的垂直漸進移動速度（正值向下，負值向上）。
摩擦力 fx	設定角色的水平移動摩擦力，使水平移動速度遞減。
摩擦力 fy	設定角色的垂直移動摩擦力，使垂直移動速度遞減。
生命週期	設定角色從建立到自動刪除的時間（單位：毫秒）。
左邊	角色的圖形左邊緣與屏幕左方的像素距離（參考下圖）。
右邊	角色的圖形右邊緣與屏幕左方的像素距離（參考下圖）。
上面	角色的圖形上邊緣與屏幕上側的像素距離（參考下圖）。
下面	角色的圖形下邊緣與屏幕上側的像素距離（參考下圖）。
z 深度	角色與鏡頭（可想像是玩家的眼睛）的距離，數值越大距離鏡頭越近，例如：A 角色深度為 1，B 角色深度為 2 時，當 A、B 在相同位置時，B 角色會蓋住 A 角色，深度值可為負值。



## 5 積木五

第五個積木是「控制器」積木盒中的「控制器移動角色」積木指令，是設定遊戲機的上、下、左、右按鈕對應於角色的移動速度及方向，vx 是水平移動的速度；vy 是垂直移動的速度，數值越大時角色的移動速度越快，如為負值則往相反方向移動。



## 6 積木六

第六個積木是「資訊」積木盒中的「生命設為」積木指令，是設定遊戲主角的生命值，遊戲主角（類別為：Player）一旦與敵人（類別為：Enemy）碰撞，生命值將自動減 1，生命值為 0 時遊戲結束，如果沒有設定生命值，預設生命值為 3。

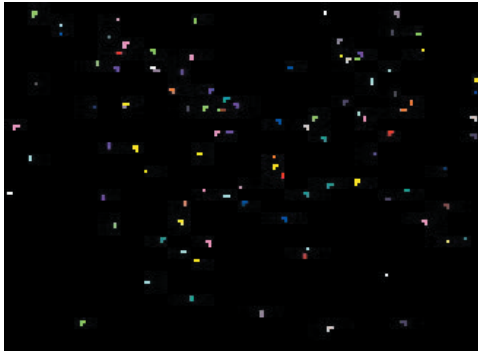


## 7 積木七

第七個積木是「場景」積木盒中的「效果 .. 用於開始畫面」積木指令，是用於設定遊戲背景效果，點選右側「+」可設定效果出現的時間，單位為毫秒。



### 七種遊戲背景效果



● 紙片效果



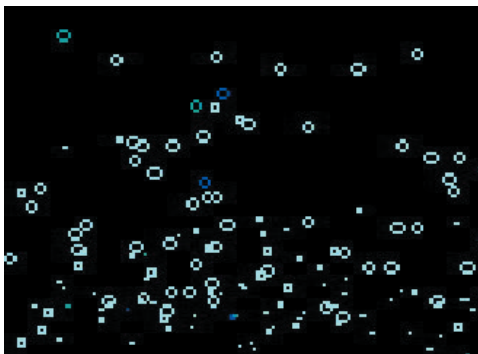
● 愛心效果



● 笑臉效果



● 暴雪效果



● 泡泡效果



● 星空效果



● 雲朵效果

## 3.4 「當 A 鍵被按住」事件積木 (按 A 鍵發射砲彈)

遊戲中控制器按下「A」鍵時，會做兩件事：

### 程式碼說明



▼行號 02  
發射砲彈時產生 0.1 秒的發射效果。

▼行號 01  
讓太空船發射砲彈來射擊隕石，以避免太空船被隕石撞毀。

### 1 積木一

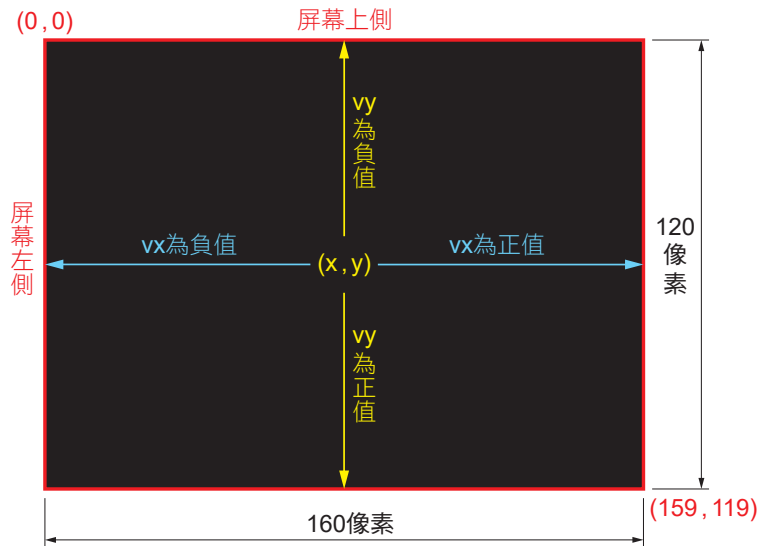
第一個積木為「角色」積木盒中的「變數設為發射物來自...」積木指令，會在屏幕上創建一個發射物，預設名稱為「projectile」，發射物預設的角色類別為「projectile」（雖然變數名稱跟類別名稱一樣，但兩者是完全不同的資料型態不可搞混，未來設計有產生發射物時應該立即將發射物變數名稱改掉避免搞混）。



發射物的第一個參數是發射物的外觀，使用圖形表示，可以點選發射物圖形後使用圖形編輯器修改或是使用素材庫中的圖形，同學可以試著改變砲彈的外型，或是增加砲彈圖片的解析度，讓太空船發射威力無比的巨大砲彈。

發射物的第二個參數是發射物來自哪個角色，因為砲彈是要從太空船發射，所以設定發射物來自太空船。如此，太空船不管在哪個位置，當按下「A」鍵，砲彈會從太空船的位置射出。

發射物的第三、四個參數是發射物的移動速度和方向，vx 是水平移動速度（正值往右，負值往左），vy 是垂直移動速度（正值往下，負值往上），速度值越大發射物移動速度越快。



為避免發射物的名稱跟發射物的角色類別搞混，我們將發射物的名稱更改為「砲彈」。



為了讓太空船更厲害，當按下「A」鍵時我們讓太空船同時發射三個方向的砲彈，現在就來增加往左上和往右上射擊的砲彈。

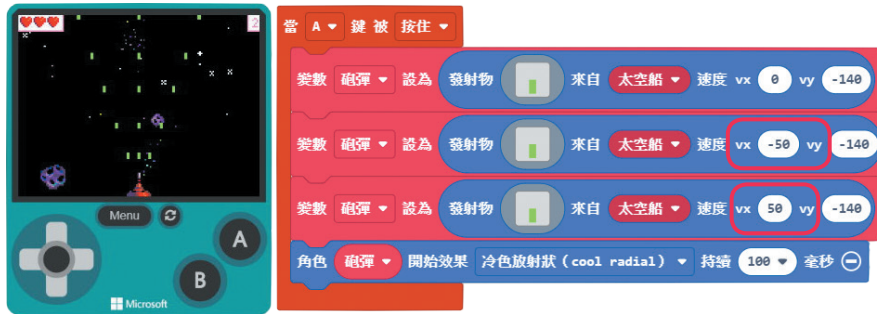
**step 1** 在砲彈發射物的積木上按滑鼠右鍵，下拉清單中點選「複製」。

**step 2** 將複製出來的積木拖曳至第一個砲彈發射物積木下方。



**step 3** 使用相同的方法複製出第三個砲彈發射物，並拖曳至第二個砲彈發射物積木下方。

**step 4** 由於下方兩個砲彈發射物要朝左上和右上發射，因此，修改 vx 速度，一個為「-50」，另一個 vx 為「50」。



## 2 積木二

第二個積木為設定效果積木，當按下「A」鍵太空船發射砲彈時產生 100 毫秒（0.1 秒）的動畫效果，同學可以試試其他不同的發射效果，也可以再複製多增加幾個其他的發射效果。



## 3.5 「當遊戲更新間隔」事件積木 (每 0.5 秒掉落隕石)

在遊戲進行過程中，遊戲內部有一個計時器同步在計時，單位為毫秒，遊戲設定每隔 500 毫秒（0.5 秒）就會觸發「當遊戲更新間隔 500 毫秒」事件一次，如此每隔 0.5 秒屏幕上面就會產生一顆隕石往下掉，在這個事件中要處理的三件事如下：

### 程式碼說明



▼行號 03  
設定隕石水平隨機位置在屏幕邊界內。

▼行號 02  
發射物原預設角色類別為「projectile」（發射物），將隕石的角色類別改為「Enemy」（敵人），以跟砲彈的「projectile」角色類別做區分。

▼行號 01  
在屏幕上方產生一顆隕石往下掉，使用隕石圖片為 asteroids(0) ~ asteroids(5) 中隨機的一張圖片，創建隕石是使用「角色」積木盒中的「變數設為發射物來自側邊」積木指令。

原始程式設定 0.5 秒產生的發射物跟按「A」鍵產生的發射物名稱一樣，為讓程式更容易理解，我們要將 0.5 秒產生的發射物名稱更換為「隕石」。

1. 在「變數設為發射物來自側邊」積木下拉變數名稱清單，選擇「新的變數」，建立一個變數名稱為「隕石」
2. 將事件內的其他二個積木的角色名稱也更改為「隕石」。



當遊戲更新 間隔 500 毫秒

變數 砲彈 設為 發射物 取得 asteroids 的項目值 索引值為 隨機取數 0 到 asteroids 的長度 - 1 來自側邊 速度 vx 0 vy 75

asteroids  
projectile  
太空船  
隕石  
新的變數...  
重新命名變數...  
刪除變數 砲彈

新變數的名稱：  
隕石  
確定

當遊戲更新 間隔 500 毫秒

變數 隕石 設為 發射物 取得 asteroids 的項目值 索引值為 隨機取數 0 到 asteroids 的長度 - 1 來自側邊 速度 vx 0 vy 75

角色 隕石 的類別設為 Projectile

set 隕石 x to 隨機取數 10 到 150

## 1 積木一

事件內的第一個積木比較複雜，我們要分析比較複雜的積木組合時，必須將積木一一拆解後再來理解：

**step 1** 首先，我們想要建立一個名為「隕石」的發射物，所以需要用到「角色」積木盒中的「變數設為發射物來自側邊」積木指令，因為隕石是要從上面垂直掉下來，所以水平速度  $vx$  設為 0；垂直速度  $vy$  設為 75。



**step 2** 這個發射物的圖片是來自於一開始就設定好的 `asteroids` 圖片陣列中 `asteroids(0)` ~ `asteroids(5)` 其中的一張圖片。因此「變數設為發射物來自側邊」積木的第一個參數，需使用「陣列」積木中的「取得陣列的項目值」積木指令，並將要取得圖片的陣列名稱更改為「`asteroids`」。



**step 3** 因為要從目前 6 張隕石圖片隨機取一張，所以使用「數學」積木盒中的「隨機取數」積木，其參數設定為數字 0 到 5，這樣「取得陣列的項目值」積木就會從「`asteroids`」陣列中 6 張圖片隨機取出其中的一張。



**step 4** 但是隨機取數時必須考慮「`asteroids`」陣列中隕石圖片有可能會增加或刪除。所以，我們先取出「`asteroids`」陣列的總數量（陣列項目的長度）然後減一，這才是隨機取數的結束值。



**step 5** 將「`asteroids`」陣列項目的總數量減一的值放入隨機取數積木的結束值中。



**step 6** 完成後的積木如下：



## 2 積木二

由於發射物的角色類別預設都是「Projectile」（發射物），為避免跟「砲彈」的角色類別相混淆，所以在第二個積木中將角色「隕石」類別設為「Enemy」（敵人）。



## 3 積木三

第三個積木設定隕石在屏幕上方的初始水平位置（x），因為屏幕寬度為 160 像素（0 ~ 159），為避免「隕石」出現在屏幕兩側的邊緣，隕石出現的水平位置左右各排除約 10 像素。

**step 1** 使用「角色」積木盒中的「設定角色屬性」積木指令

**step 2** 水平位置（x）使用「數學」積木盒中的「隨機取數」積木指令，參數設定從 10 到 150。



## 3.6 「碰撞偵測」事件積木（偵測太空船和隕石碰撞）

由於 Arcade 的物理碰撞偵測是採用某角色類別是否與另一角色類別重疊，所以，在角色建立的時候就應該將角色的類別定義清楚，在這個遊戲中各角色的類別如下：

角色	類別
太空船	Player（預設值）
砲彈	Projectile（預設值）
隕石	Enemy

在這個事件積木是要偵測太空船（角色類別：Player）是否碰撞到隕石（角色類別：Enemy），如果有碰撞則生命值要減一。

特別注意：因為 Arcade 碰撞偵測是使用「重疊角色」積木，所以當重疊事件觸發後必須至少將其中一個角色刪除，否則如果兩個角色都還繼續存在屏幕上，將會繼續觸發重疊事件，事件內的程式積木也將會被執行多次。

在這個事件中要處理的四件事如下：

### 程式碼說明

#### ▼行號 01

碰撞時讓玩家感受到太空船受創的慘烈畫面。

#### ▼行號 02

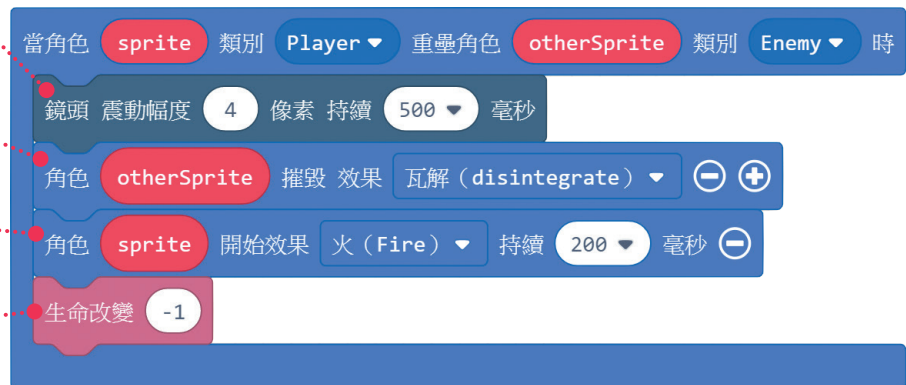
刪除隕石並有瓦解的特效。

#### ▼行號 03

讓太空船產生火光效果。

#### ▼行號 04

生命值減一，當生命值為 0 時遊戲結束。



### ● 偵測太空船和隕石碰撞

## 1 積木一

事件中的第一個積木使用的是「場景」積木盒中的「鏡頭震動」積木，可以讓玩家感受到太空船受創的慘烈畫面，其中震動幅度為讓整個屏幕畫面暫時移動的像素距離，為免影響遊戲進行，震動持續時間不宜太久。另外同學可以從「音效」積木盒中取出「播放音效」積木，讓太空船撞到隕石的同時具有聲光效果。

鏡頭 震動幅度 4 像素 持續 500 毫秒

## 2 積木二

第二個積木是要將觸發「重疊角色」事件的「隕石」角色刪除，並且產生「瓦解」的動畫效果。要特別注意的是在「重疊角色」事件中要控制角色的行為（例如刪除角色），不能使用角色名稱或角色類別，以刪除「隕石」角色為例，整個屏幕上有很多名稱都叫做「隕石」的角色，角色類別也都叫做「Enemy」，但我們真正要刪除的是觸發「重疊角色」事件的那顆「隕石」，所以就必須使用事件積木上「Enemy」類別前面的那個角色代稱「otherSprite」來操作「角色摧毀」。



## 3 積木三

同樣的第三個積木是要讓觸發「重疊角色」事件的「太空船」角色產生火光的效果，所以就必須使用事件積木上「Player」類別前面的那個角色代稱「Sprite」來操作角色的「開始效果」。



## 4 積木四

第四個積木使用「資訊」積木盒中的「生命改變」積木，讓玩家生命值減一，生命值會出現在屏幕左上角，如果生命值為 0，遊戲將自動結束，屏幕出現「GAME OVER」。



## 3.7 「碰撞偵測」事件積木（偵測砲彈和隕石碰撞）

在這個事件積木是要偵測砲彈（角色類別：Projectile）是否碰撞到隕石（角色類別 Enemy），如果有碰撞則得分值要加一。

特別注意：因為 Arcade 碰撞偵測是使用「重疊角色」積木，所以當重疊事件觸發後必須至少將其中一個角色刪除，否則如果兩個角色都還繼續存在屏幕上，將會繼續觸發重疊事件，事件內的程式積木也將會被執行多次，一般而言一顆砲彈只會擊中一顆隕石，所以，事件中的砲彈和隕石都必須刪除。

在這個事件中要處理的三件事如下：

### 程式碼說明

▼行號 01  
刪除砲彈。

▼行號 02  
刪除隕石並有瓦解的特效。

▼行號 03  
得分加一。

當角色 **sprite** 類別 **Projectile** 重疊角色 **otherSprite** 類別 **Enemy** 時

- 角色 **sprite** 摧毀 +
- 角色 **otherSprite** 摧毀 效果 瓦解 (disintegrate) - +
- 得分改變 1

### 偵測砲彈和隕石碰撞

## 1 積木一

事件中的第一個積木是要將觸發「重疊角色」事件的「砲彈」角色刪除，所以使用事件積木「砲彈」的類別「Projectile」前面的那個角色代稱「Sprite」來操作「角色摧毀」。



## 2 積木二

第二個積木是要將觸發「重疊角色」事件的「隕石」角色刪除，所以使用事件積木「Enemy」類別前面的那個角色代稱「otherSprite」來操作「角色摧毀」並且產生「瓦解」效果。



## 3 積木三

第三個積木使用「資訊」積木盒中的「得分改變」積木，讓玩家得分加一，分數會出現在屏幕右上角。



## 3.8 本章使用到 Arcade 遊戲設計相關名詞

名詞	說明
事件觸發	Arcade 程式積木都必須放在事件積木中，當該事件被觸發時，內部的積木程式才會被執行。
角色	角色為屏幕上可獨立移動的圖像，建立角色時除了設定對應的角色名稱外，也會預設一個角色類別。
角色類別	在創建遊戲主角時其角色類別預設為「Player」，創建發射物角色時其角色類別預設為「Projectile」，角色類別可自行定義。
發射物	創建發射物時發射物出現的位置有兩種，一種是從玩家角色的位置出現；另一種是從屏幕邊緣出現，設定發射物的速度 $v_x$ 和 $v_y$ 可設定其移動方向。
重疊角色	Arcade 採用的碰撞偵測是某一角色類別重疊到另一角色類別時，會觸發「重疊角色」事件，在事件中必須將其中一個角色刪除，否則會持續觸發「重疊角色」事件，在「重疊角色」事件中要操作觸發角色的行為必須使用類別前面的角色代稱（Sprite 或 otherSprite）來操作。
角色摧毀	「發射物」的角色一旦離開屏幕範圍將自動銷毀。其他在遊戲中創建的角色將一直保留在遊戲中，直到使用「角色摧毀」積木或設定角色的「生命週期」。
位置	Arcade 遊戲機的預設屏幕解析度為 $160 \times 120$ 像素，左上角座標為 (0,0)，右下角座標為 (159,119)。
速度	角色的移動速度分為速度和加速度，速度是等速運動，加速度為漸進加快， $v_x$ 或 $a_x$ 正值時角色往右移動（負值時相反）， $v_y$ 或 $a_y$ 正值時角色往下移動（負值時相反）
素材庫	Arcade 編輯器內建一些貼圖素材，可以用來設定為角色圖示或背景圖案。

## 3.9 修改後完整程式碼

〔程式檔：03 太空驅逐艦〕

The image displays two sections of Scratch code blocks. The top section is titled "當啟動時" (When the game starts) and includes the following blocks: a "陣列" (Array) block containing images of a burger, strawberry, lemon, orange, pink candy, and pizza; a "變數 asteroids 設為" (Set variable asteroids to) block; a "變數 太空船 設為 角色" (Set variable spaceship to character) block with a spaceship icon and "類別 Player" (Category Player); a "角色 太空船 標記 留在畫面 設為" (Character spaceship mark stay on screen set to) block with a "是" (Yes) button; a "set 太空船 下面 to 120" (Set spaceship bottom to 120) block; a "控制器移動角色 太空船 速度 vx 100 vy 100" (Controller move character spaceship speed vx 100 vy 100) block; a "生命設為 3" (Set life to 3) block; and an "效果 星空 (star field) 用於開始畫面" (Effect star field for start screen) block. The bottom section is titled "當遊戲更新 間隔 500 毫秒" (When the game updates every 500 milliseconds) and includes: a "變數 隕石 設為 發射物 取得 asteroids 的項目值 索引值為 隨機取數 0 到 asteroids 的長度 - 1 來自側邊 速度 vx 0 vy 75" (Set variable meteor to projectile get asteroids item value index value random number 0 to asteroids length - 1 from side speed vx 0 vy 75) block; a "角色 隕石 的類別設為 Enemy" (Character meteor category set to Enemy) block; and a "set 隕石 x to 隨機取數 10 到 150" (Set meteor x to random number 10 to 150) block.

程式碼續接下頁

```

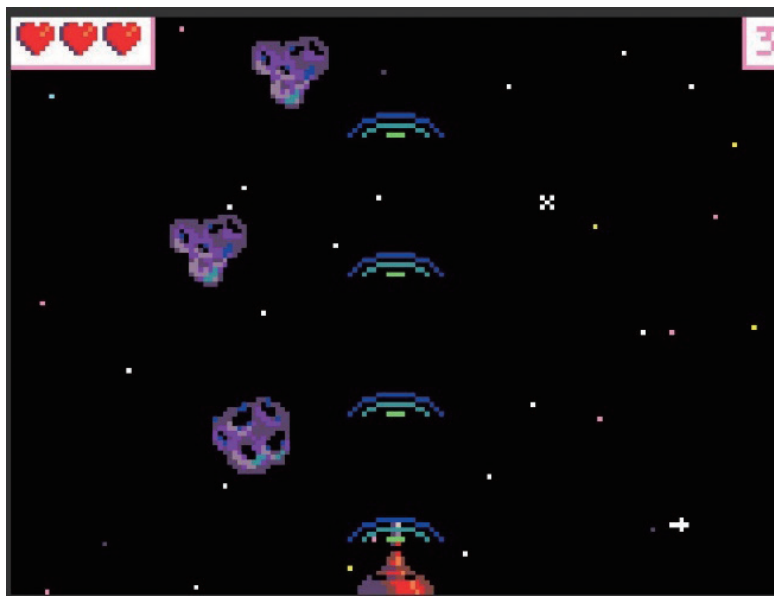
當 A 鍵被按住
    變數 砲彈 設為 發射物 來自 太空船 速度 vx 0 vy -140
    變數 砲彈 設為 發射物 來自 太空船 速度 vx -50 vy -140
    變數 砲彈 設為 發射物 來自 太空船 速度 vx 50 vy -140
    角色 砲彈 開始效果 冷色放射狀 (cool radial) 持續 100 毫秒

當角色 sprite 類別 Projectile 重疊角色 otherSprite 類別 Enemy 時
    角色 sprite 摧毀
    角色 otherSprite 摧毀 效果 瓦解 (disintegrate)
    得分改變 1

當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Enemy 時
    鏡頭 震動幅度 4 像素 持續 500 毫秒
    角色 otherSprite 摧毀 效果 瓦解 (disintegrate)
    角色 sprite 開始效果 火 (Fire) 持續 200 毫秒
    生命改變 -1
    
```

### 練習一下

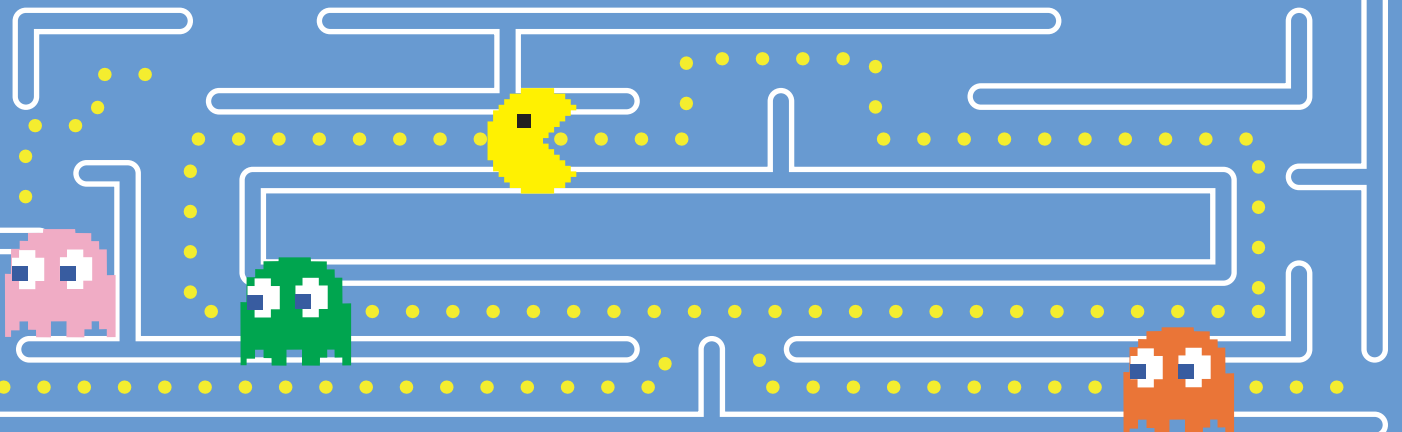
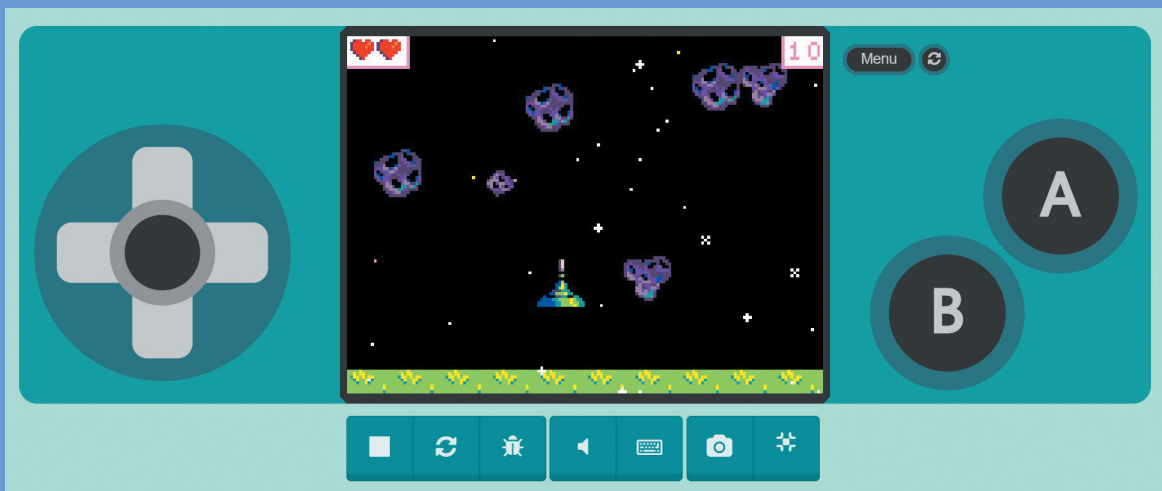
增加當 B 鍵按下時，發射音波彈，如下圖所示：



# 4 Chapter

## 「收集隕石任務」遊戲

本章將學習繪製遊戲背景；圖片陣列設定與取用；角色和角色、角色和貼圖間的碰撞偵測及處理；陣列圖片的設定及取得；依照得分來設定時間延遲等應用。





## 遊戲說明

為了保護地球免於遭受天外的隕石襲擊，太空中心設計了一艘專門處理隕石的太空船，要在隕石掉落到地球前將隕石收集起來，地球一旦遭受到三顆隕石撞擊，地球將毀滅遊戲也將結束。

## 4.1 遊戲設計分析

### 一、遊戲角色：

角色名稱	角色類別	素材	創建角色的事件
太空船	Player		當啟動時，初始位置（100,100）
隕石群（0～3）	Food		重複無限次

### 二、遊戲事件：

#### 收集隕石任務

當啟動時(產生遊戲主角太空船、繪製背景，設定生命值。)

重複無限次(產生隕石發射物，隨機出現在屏幕上方，並依「延時」變數內容，更改隕石產生速度。)

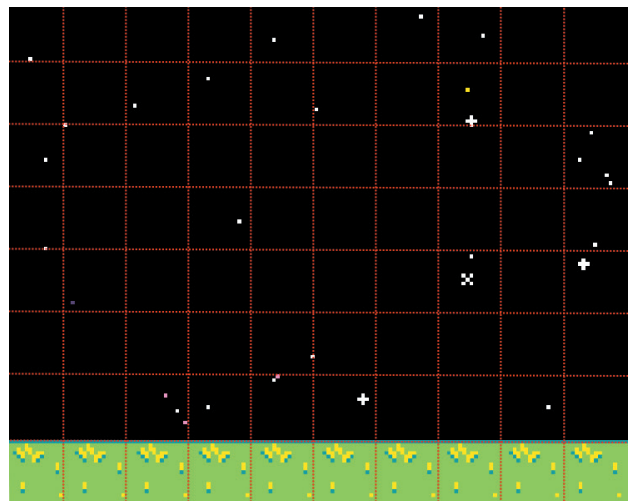
當遊戲更新(依照得分來設定「延時」變數內容，以此改變隕石產生的速度。)

角色類別重疊(偵測太空船和隕石是否碰撞，如果碰撞得分+1，該隕石刪除。)

角色類別重疊貼圖(偵測隕石是否碰撞地面，如果碰撞生命值-1，該隕石刪除。)

### 三、背景設計：

為限制遊戲範圍不超過屏幕範圍，貼圖地圖為  $10 \times 8$  塊貼圖，背景效果為「星空動畫」，貼圖地圖下方放置草地貼圖代表地球，如右圖所示：



## 4.2 「當啟動時」事件


遊戲一開始時屏幕顯示背景、創建遊戲角色及設定角色物理動作、生命值等。

### 程式碼說明















▼行號 01  
當啟動時執行事件區塊內的程式


▼行號 03  
設定遊戲畫面為「星空」效果


當啟動時


貼圖地圖設為 

效果 星空 (star field) 用於開始畫面

變數 隕石群 設為 陣列               

變數 太空船 設為 角色  類別 Player

角色 太空船 標記 留在畫面 設為 

控制器移動角色 太空船 速度 vx 200 vy 200 

生命設為 3

▼行號 04  
創建「隕石群」陣列變數

① 使用「進階／陣列」積木盒中的「變數設為陣列」積木指令（數字或文字陣列都可以），將變數名稱改為「隕石群」

② 按「+」新增二個空白項目，使得陣列清單有四個項目內容，分別放置「圖像」積木盒中的「一個圖像」（漢堡圖示）積木指令，並下拉圖像選單，依序選擇四種不同的隕石圖示

▼行號 05  
創建遊戲主角「太空船」

▼行號 06  
設定「太空船」不能移出屏幕外

▼行號 07  
設定控制器與角色「太空船」做連結，能使用方向鍵移動「太空船」在屏幕中的位置

▼行號 08  
將玩家生命值設為 3

▼行號 02  
繪製遊戲背景。

① 使用「場景」積木盒中的「貼圖地圖設為..」積木指令，點選空白貼圖開啟「地圖編輯器」

② 繪製草地

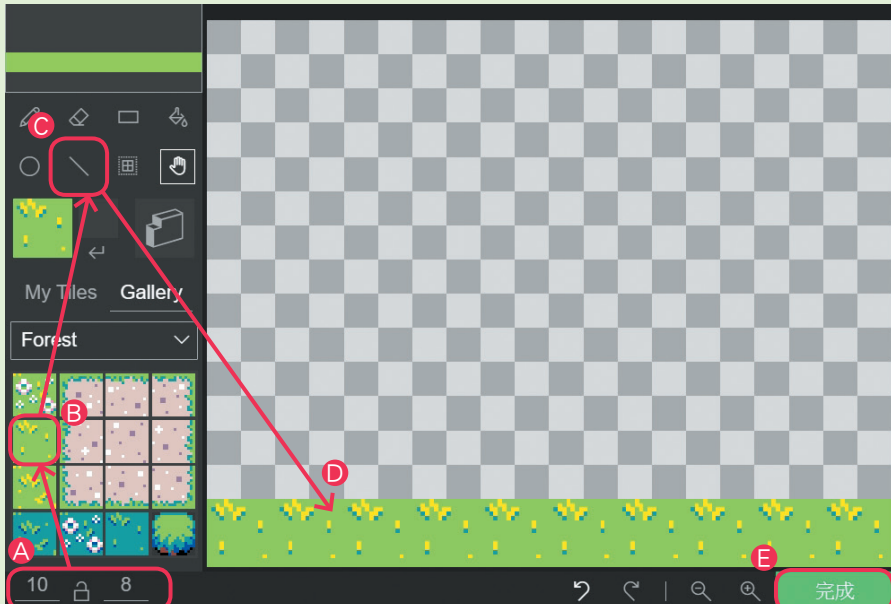
A 設定地圖解析度為「10×8」塊貼圖

B 點選「草地」貼圖

C 選擇「畫線」工具

D 地圖下方繪製「草地」貼圖

E 按「完成」



## 4.3 「重複無限次」事件

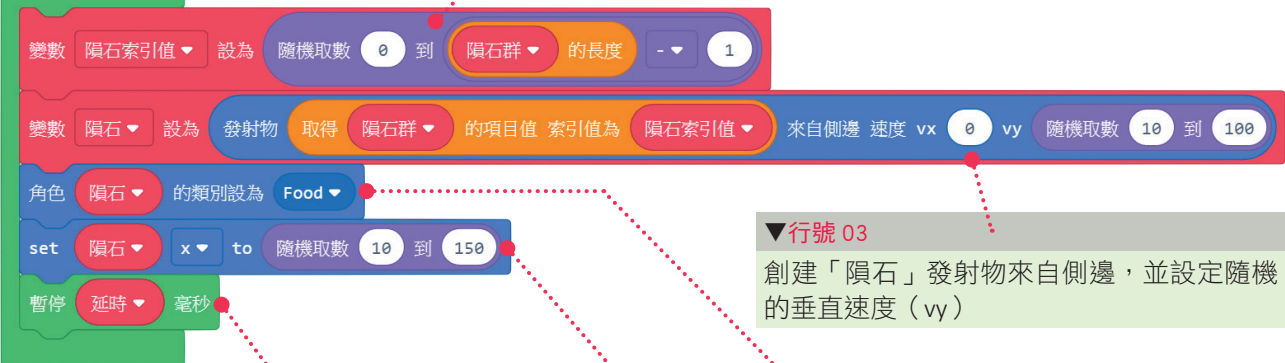
遊戲執行中隕石將不斷的從屏幕上方隨機的位置往下掉隕石，隕石掉下的速度也是隨機的，以此增加遊戲的難度，並且間隔一段時間才會繼續產生下一顆隕石（因事件中有使用到暫停積木指令，因此避免使用在「當遊戲更新」事件）。

### 程式碼說明

#### ▼行號 01

當遊戲進行中無限次的重複執行事件區塊內的程式

重複無限次



The code block consists of the following blocks from top to bottom:

- Repeat block:** A red 'Repeat' block with 'Repeat' set to 'Forever' and 'Number of repeats' set to '1'. It contains:
  - Set block:** 'Variable: Meteor Index' set to 'Random number from 0 to Meteor Array length - 1'.
  - Set block:** 'Variable: Meteor' set to 'Projectile' block with 'Get item value at index Meteor Index from Meteor Array' and 'Velocity vx: 0, vy: Random number from 10 to 100'.
  - Block:** 'Character: Meteor' with 'Category set to Food'.
  - Set block:** 'set Meteor x to Random number from 10 to 150'.
  - Pause block:** 'Pause' with 'Delay: milliseconds'.

#### ▼行號 02

隨機產生一個數值（隕石索引值），做為「隕石群」陣列中要取出的隕石編號（隨機取數從「0」到「隕石數量 -1」，因為隕石是從 0 開始編號，所以，最後一顆隕石的編號為隕石數量 -1）

#### ▼行號 03

創建「隕石」發射物來自側邊，並設定隨機的垂直速度（vy）

#### ▼行號 04

原發射物預設類別為「Projectile」，更改發射物角色類別為「Food」

#### ▼行號 05

因為屏幕寬度為 160 像素，「隕石」水平設置於 10 ~ 150 的位置

#### ▼行號 06

讓下一個隕石延後一段時間再出現（此「延時」變數內容將因得分增加而減少，讓遊戲更加刺激）

## 4.4 「當遊戲更新」事件

當遊戲運行中持續不斷的檢查得分，並依得分改變隕石產生的時間，得分越高，隕石產生速度越快。

### 程式碼說明

#### ▼行號 01

當遊戲進行時重複執行事件區塊內的程式

#### ▼行號 02

依得分高低設定隕石產生速度，使用「如果」積木指令做連續的條件判斷，按「+」增加「否則如果」區塊。第一個條件設為當得分大於 20 時，則執行第一個區塊內的程式，並略過後面的區塊

#### ▼行號 03

當得分超過 20 分時每隔 0.5 秒產生隕石

#### ▼行號 04

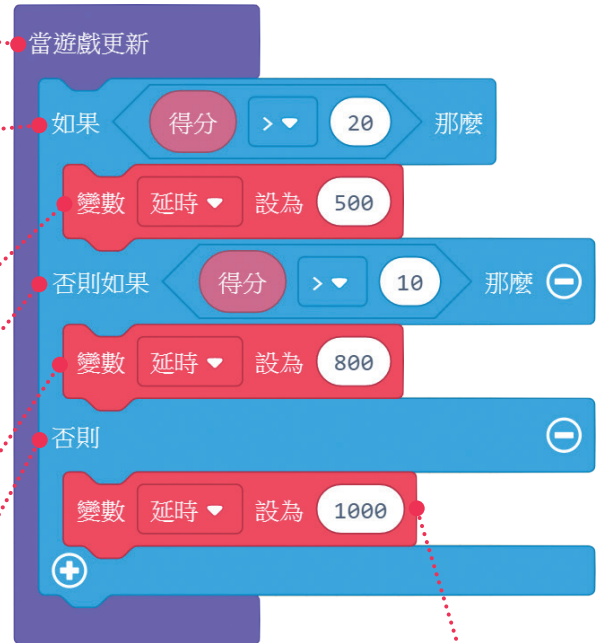
第二個條件設為當得分大於 10 時，則執行第二個區塊內的程式，並略過後面的區塊

#### ▼行號 05

當得分超過 10 分時每隔 0.8 秒產生隕石

#### ▼行號 06

當前面兩個條件都不成立時，則執行下面區塊內的程式



#### ▼行號 07

當得分未超過 10 分時每隔 1 秒產生隕石

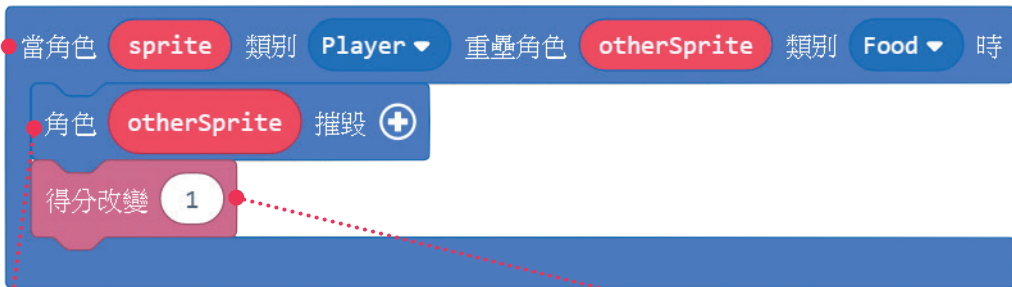
## 4.5 「角色類別重疊」事件

當太空船類別（Player）重疊隕石類別（Food）時，表示太空船蒐集到隕石，得分加 1，為避免重複計算得分，必須將與玩家重疊的隕石角色刪除，刪除角色要使用角色代稱（otherSprite），不能直接使用角色名稱，否則屏幕上相同名稱的角色都會被刪除。

### 程式碼說明

#### ▼行號 01

使用「角色」積木盒中的「角色類別重疊」事件積木，第二個類別參數下拉選擇「Food」（隕石的類別為 Food）



#### ▼行號 02

為避免重複計算得分，必須將與玩家重疊的隕石角色刪除（將「角色類別重疊」事件積木中類別為「Food」前的角色代稱「otherSprite」拖曳至「角色摧毀」積木指令參數中）

#### ▼行號 03

玩家得分加 1 分

## 4.6 「角色類別重疊貼圖」事件

當隕石（類別為 Food）碰到草地貼圖時，將該隕石摧毀並將生命值 -1，並產生音效及震動特效。

### 程式碼說明

#### ▼行號 01

當「Food」類別重疊草地貼圖時執行事件區塊內的程式



#### ▼行號 02

將玩家生命值 -1（目前軟體版本有時會出現 -2 情形）

#### ▼行號 03

產生畫面震動特效

#### ▼行號 04

播放撞擊地面音效

#### ▼行號 05

為避免生命值重複減一，必須將與貼圖重疊的隕石角色刪除（將「角色類別重疊貼圖」事件積木中類別為「Food」的角色代稱「sprite」拖曳至「角色摧毀」積木指令參數中）

## 4.7 完整程式碼

〔程式檔：04 收集隕石任務〕

**當啟動時**

- 貼圖地圖設為 [ ]
- 效果 星空 (star field) 用於開始畫面
- 變數 隕石群 設為 陣列 [隕石] [隕石] [隕石] [隕石]
- 變數 太空船 設為 角色 [太空船] 類別 Player
- 角色 太空船 標記 留在畫面 設為 [ ]
- 控制器移動角色 太空船 速度 vx 200 vy 200
- 生命設為 3

**當角色 sprite 類別 Food 重疊貼圖 [ ] 位置 location**

- 生命改變 -1
- 鏡頭 震動幅度 4 像素 持續 500 毫秒
- 播放音效 power down
- 角色 sprite 摧毀

**重複無限次**

- 變數 隕石索引值 設為 隨機取數 0 到 隕石群 的長度 - 1
- 變數 隕石 設為 發射物 取得 隕石群 的項目值 索引值為 隕石索引值 來自側邊 速度 vx 0 vy 隨機取數 10 到 100
- 角色 隕石 的類別設為 Food
- set 隕石 x to 隨機取數 10 到 150
- 暫停 延時 毫秒

**當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Food 時**

- 角色 otherSprite 摧毀
- 得分改變 1

**當遊戲更新**

- 如果 得分 > 20 那麼
  - 變數 延時 設為 500
- 否則如果 得分 > 10 那麼
  - 變數 延時 設為 800
- 否則
  - 變數 延時 設為 1000

### 練習一下

請修改「當遊戲更新」事件內的程式，並依以下表格條件設定隕石產生的速度。

「得分」變數內容	「延時」變數內容
0 ~ 10 分	1000ms
11 ~ 20 分	800ms
21 ~ 30 分	600ms
31 分 (含) 以上	400ms

# 綜合實作題

**題目名稱** 射擊隕石遊戲

**題目說明** 請修改本章「收集隕石任務」遊戲，將生命值增加為「10」，遊戲改成按下「A」鈕時，太空船向上發射砲彈，利用砲彈來擊毀隕石，並且得分「+1」。

**解題提示**

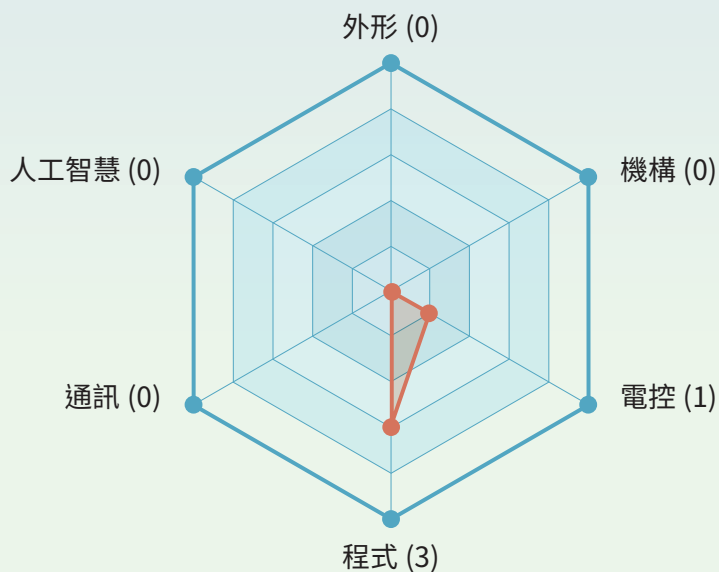
新增「當 A 鍵被重複」事件，並在事件中加入一個名為砲彈的「發射物來自…」積木指令；在「當角色類別重疊」事件中，更改「sprint」後面的類別為「projectile」（發射物的預設類別）。

創客題目編號：A033002

創客指標

外形	0
機構	0
電控	1
程式	3
通訊	0
人工智慧	0
<b>創客總數</b>	<b>4</b>

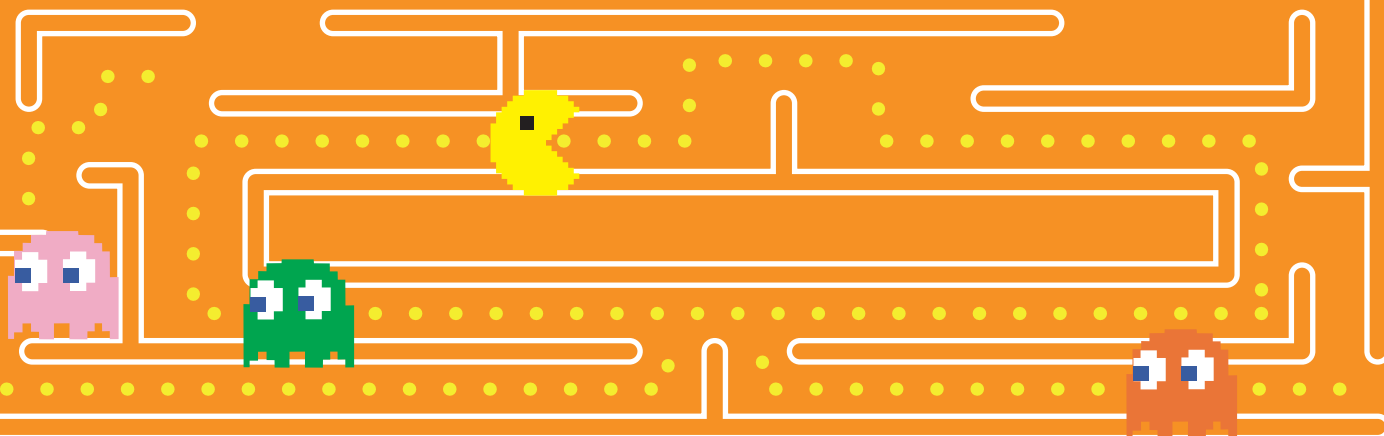
45  
mins



# 5 Chapter

## 「離線小鴨鴨」遊戲

本章模擬 Chrome 瀏覽器網路斷線時，經典的離線小恐龍遊戲，學習的重點在於練習橫軸遊戲中無限循環的地板；建立自訂函式；角色往上跳躍與地心引力之間的作用；障礙物的出現等。




## 遊戲說明

小鴨走在有很多木頭障礙物的地面上，小鴨必須安全的跳過木頭才能繼續往前走，小鴨走的越遠分數將越高，一旦撞到障礙物遊戲將結束。

## 5.1 遊戲設計分析

### 一、遊戲角色：

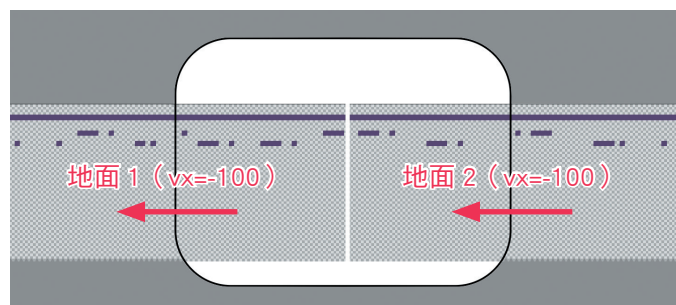
角色名稱	角色類別	素材	創建角色的事件
小鴨	Player		當啟動時，初始位置 (10,100)
木頭	發射物 (Projectile)		當遊戲更新間隔 1000ms(60%)

### 二、遊戲事件：



### 三、背景設計：

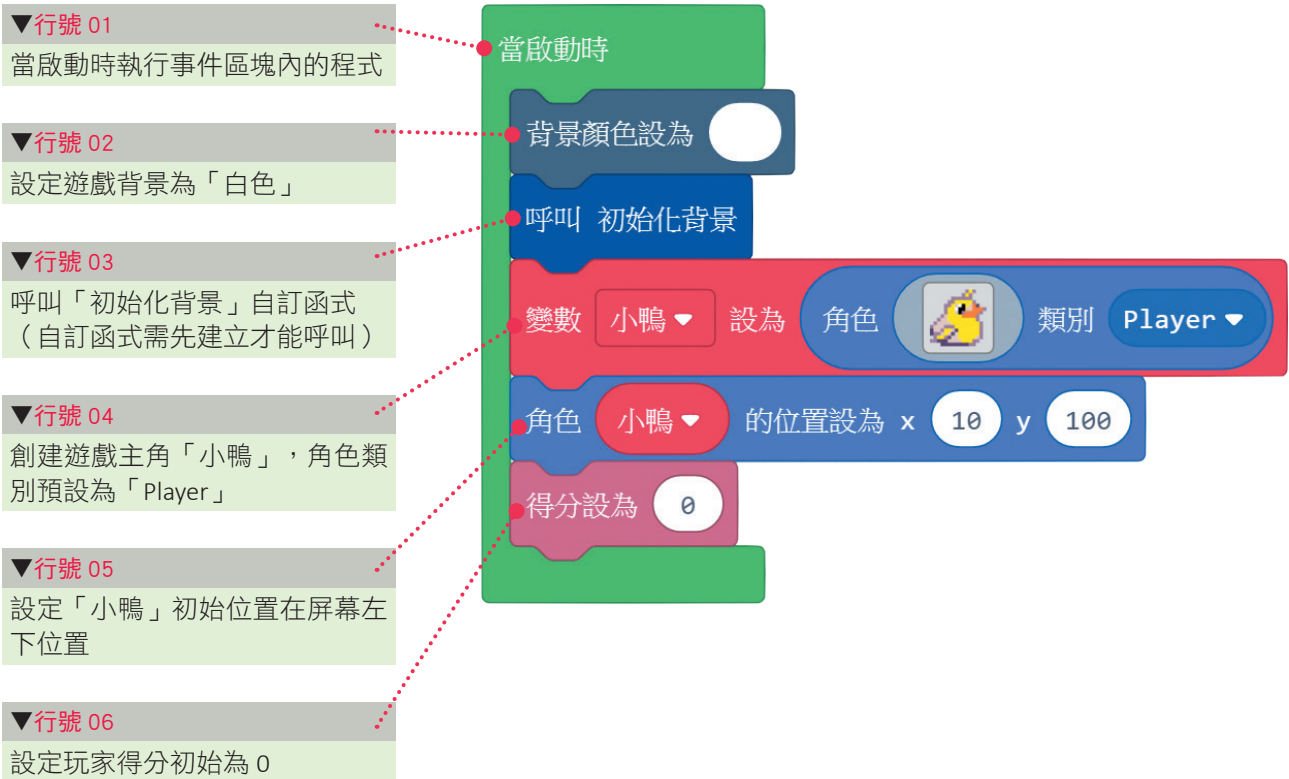
因為地面要有橫向捲動的效果，因此地面不能使用背景貼圖（背景貼圖無法移動），因此創建二個相同的地面角色（圖像大小 162×30 像素），遊戲中再讓二個地面輪流在屏幕上移動，形成地面向前的動畫，如下圖所示：



## 5.2 「當啟動時」事件

遊戲一開始時屏幕顯示背景、創建遊戲角色及設定角色物理動作等。

### 程式碼說明



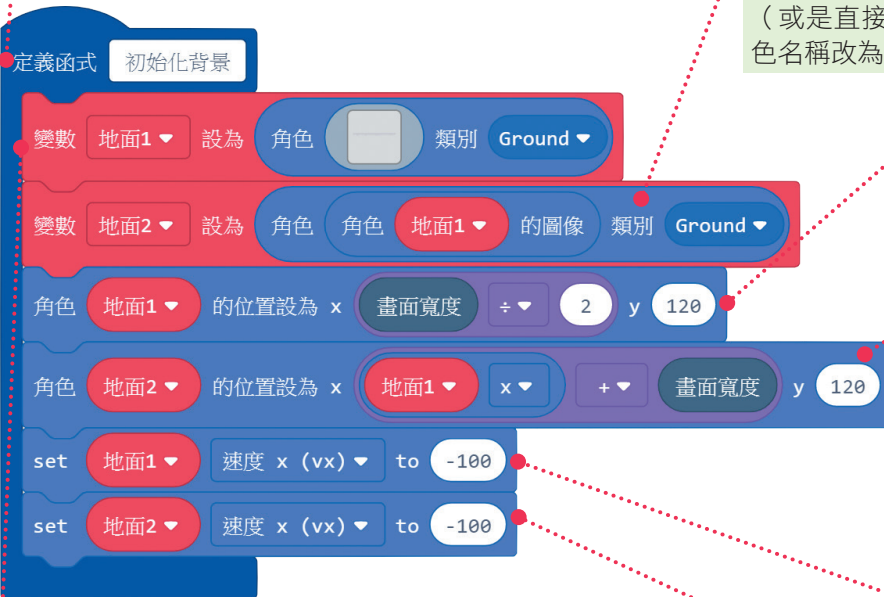
「初始化背景」自訂函式：

新增二個地面角色，寬度均為 162 像素；高度均為 30 像素，「地面 1」的中心點水平 (x) 初始位置在 80 (畫面寬度 ÷ 2)，垂直 (y) 位置在「鴨子」的下方 (120)，「地面 2」水平初始位置緊接在「地面 1」的右側 (80 + 160)，垂直位置和「地面 1」相同 (120)。

## 程式碼說明

## ▼行號 01

當程式中呼叫自訂函式時執行函式區塊內的程式



定義函式 初始化背景

變數 地面1 設為 角色 類別 Ground

變數 地面2 設為 角色 角色 地面1 的圖像 類別 Ground

角色 地面1 的位置設為 x 畫面寬度 ÷ 2 y 120

角色 地面2 的位置設為 x 地面1 x + 畫面寬度 y 120

set 地面1 速度 x (vx) to -100

set 地面2 速度 x (vx) to -100

## ▼行號 03

創建遊戲角色「地面 2」，其圖像跟「地面 1」相同即可  
(或是直接複製「地面 1」的積木，再將角色名稱改為「地面 2」即可)

## ▼行號 04

設定「地面 1」水平中心位置 (x) 於屏幕的中心點 (畫面寬度 ÷ 2)，垂直位置為 120

## ▼行號 05

設定「地面 2」緊接在「地面 1」的右側 (「地面 1」水平中心點 + 畫面寬度)，垂直位置為 120  
(或參考使用「重複無限次」事件中的地面定位方式)

## ▼行號 06

設定「地面 1」初始由右往左移動 (vx=-100)

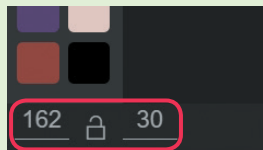
## ▼行號 07

設定「地面 2」，初始由右往左移動 (vx=-100)

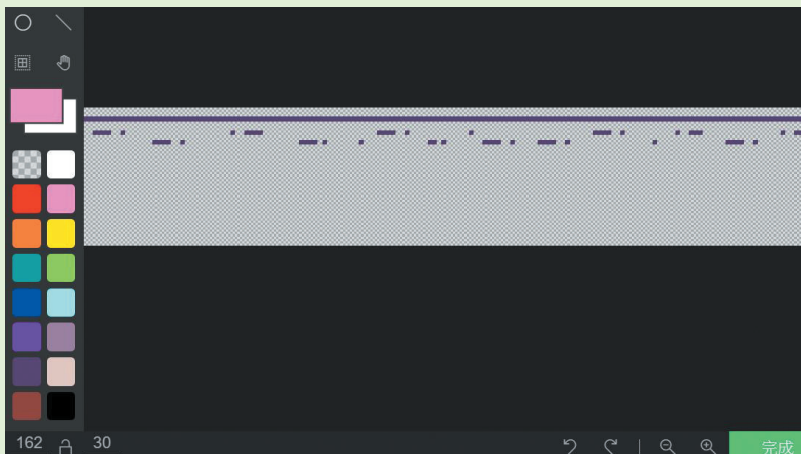
## ▼行號 02

創建遊戲角色「地面 1」

- ① 使用「角色」積木盒中的「變數設為角色」積木指令，變數名稱改為「地面 1」
- ② 下拉類別選單，選擇「添加新的 kind」，輸入自訂的「Ground」類別
- ③ 點選空白角色貼圖，進入貼圖編輯器
- ④ 貼圖編輯器左下解析度更改為「162×30」(屏幕寬度為 160)



- ⑤ 在貼圖編輯區中繪製地面，如下圖所示：



## 5.3 「當任何鍵被按住時」事件

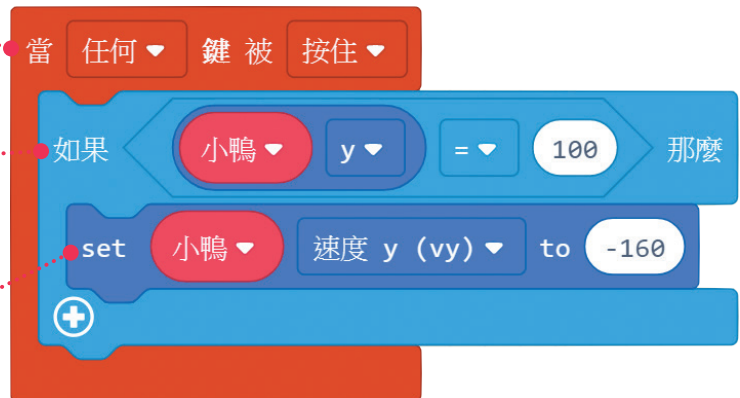
使用任何按鈕來控制主角「小鴨」往上跳，但是要限制小鴨是在地面時才能往上跳，往上跳是讓小鴨垂直往上移動。

### 程式碼說明

▼行號 01  
當任何按鈕「按住」時執行事件區塊內的程式

▼行號 02  
要限制小鴨是在地面時才能往上跳，判斷小鴨是否在地面（ $y=100$ ），如果條件成立則執行「那麼」區塊內的程式

▼行號 03  
讓小鴨垂直往上移動（ $vy=-160$ ）



## 5.4 「當遊戲更新」事件

當小鴨往上跳後需讓小鴨受到地心引力影響往下掉，因此，使用加速度的方式讓小鴨往下掉，加速度是時間越久速度越快，剛開始速度大於加速度，小鴨往上，加速度持續增加，當速度小於加速度時，小鴨開始往下掉，當小鴨已經低於地面時，設定小鴨返回原始位置，垂直加速度（ $ay$ ）將自動為 0。

### 程式碼說明

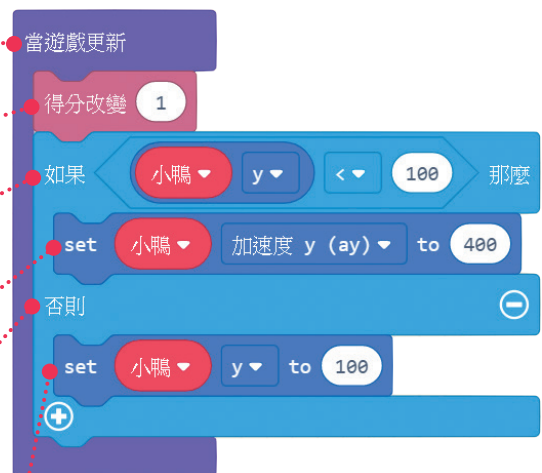
▼行號 01  
當遊戲進行時重複執行事件區塊內的程式（此事件中應避免使用延遲指令）

▼行號 02  
讓玩家得分加 1

▼行號 03  
判斷小鴨是否還在空中（ $y < 100$ ），如果是則執行「那麼」區塊內的程式

▼行號 04  
持續將小鴨往下拉，設定小鴨垂直加速度為 400

▼行號 05  
如果條件不成立（小鴨在地面）則執行「否則」區塊內的程式



▼行號 06  
設定小鴨返回原始位置，小鴨垂直位置為 100（ $y=100$ ）

## 5.5 「當遊戲更新 1000 毫秒」事件

每隔 1000 毫秒（1 秒）地面往左的移動速度將加快 1 個像素，並在屏幕右側產生「木頭」，設定往左移動（移動速度與地面一致）。

### 程式碼說明

#### ▼行號 01

當每隔 1 秒時執行事件區塊內的程式

當遊戲更新 間隔 1000 毫秒

change 地面1 速度 x (vx) by -1

change 地面2 速度 x (vx) by -1

如果 隨機取布林值 true 的機率設為 60 % 那麼

變數 木頭 設為 發射物 來自側邊 速度 vx 地面1 速度 x (vx) vy 0

set 木頭 y to 100

#### ▼行號 02

讓「地面 1」移動速度往左加快 1 個像素

#### ▼行號 03

讓「地面 2」移動速度往左加快 1 個像素

#### ▼行號 04

每隔 1 秒有 60% 的機率下執行「那麼」區塊內的產生障礙物程式

#### ▼行號 05

創建「木頭」發射物，從右側往左移動，水平移動速度設為與「地面 1」的水平速度（vx）相同


#### ▼行號 06

設定「木頭」發射物的垂直位置在地面（y=100）

## 5.6 「重複無限次」事件

遊戲執行期間使用「地面 1」和「地面 2」二個角色，讓地面由右往左產生流動效果，模擬小鴨持續往前進，在事件中主要判斷二個地面角色，如果有一地面角色已經移出屏幕，就將該地面角色設定到另一地面角色的右側。

### 程式碼說明

<p>▼行號 01 遊戲進行中持續重複執行事件區塊內的程式</p> <p>▼行號 02 判斷「地面 1」是否已移出屏幕（右側小於 0）</p> <p>▼行號 03 將「地面 1」角色移動至「地面 2」角色的右邊</p> <p>▼行號 04 判斷「地面 2」是否已移出屏幕（右側小於 0）</p> <p>▼行號 05 將「地面 2」角色移動至「地面 1」角色的右邊</p>	
---	---

## 5.7 「當類別 Player 重疊類別 Projectile」事件

當小鴨來不及跳起來或太早跳，導致撞到木頭時遊戲結束。


### 程式碼說明

	
	
<p>▼行號 01 當玩家類別碰到發射物類別時執行事件區塊內的程式</p>	<p>▼行號 02 屏幕顯示「GAME OVER」遊戲結束</p>

## 5.8 完整程式碼

〔程式檔：05 離線小鴨鴨〕


**當啟動時**

- 背景顏色設為
- 呼叫 初始化背景
- 變數 小鴨 設為 角色  類別 Player
- 角色 小鴨 的位置設為 x  y
- 得分設為

**重複無限次**

- 如果  右邊  那麼
- set  左邊 to  右邊
- +
- 如果  右邊  那麼
- set  左邊 to  右邊
- +

**定義函式 初始化背景**

- 變數 地面1 設為 角色  類別 Ground
- 變數 地面2 設為 角色   的圖像 類別 Ground
- 角色 地面1 的位置設為 x  ÷  y
- 角色 地面2 的位置設為 x  x +  y
- set  速度 x (vx) to
- set  速度 x (vx) to

**當 任何 鍵 被 按住**

- 如果  y  那麼
- set  速度 y (vy) to
- +

**當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Projectile 時**

- 遊戲結束 結果
- +

**當遊戲更新**

- 得分改變
- 如果  y  那麼
- set  加速度 y (ay) to
- 否則
- set  加速度 y (ay) to
- set  y to
- +

**當遊戲更新 間隔 1000 毫秒**

- change  速度 x (vx) by
- change  速度 x (vx) by
- 如果 隨機取布林值 true 的機率設為  % 那麼
- 變數 木頭 設為 發射物  來自側邊 速度 vx  速度 x (vx) vy
- set  y to
- +

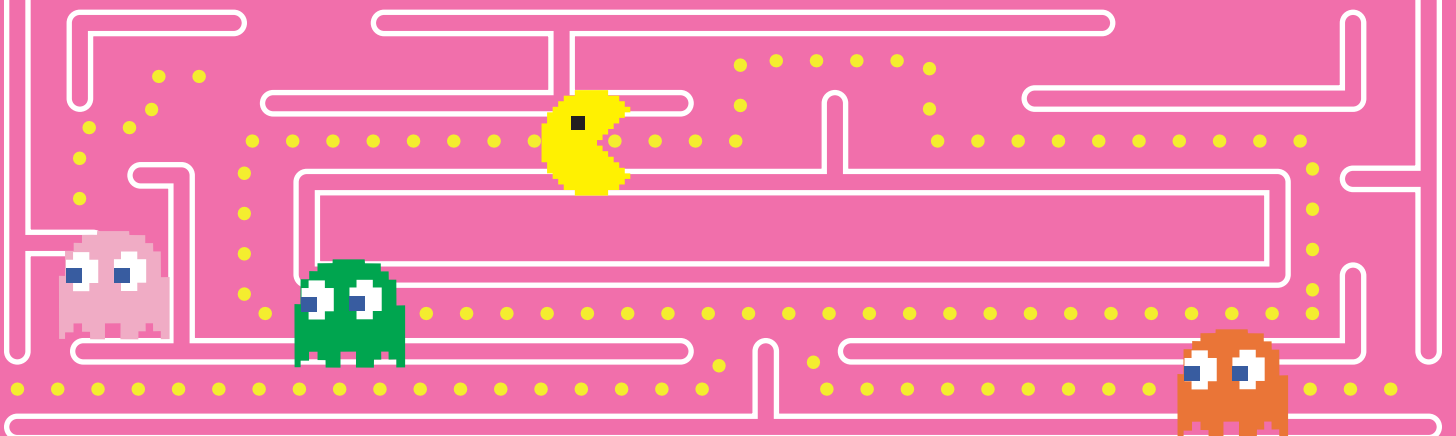
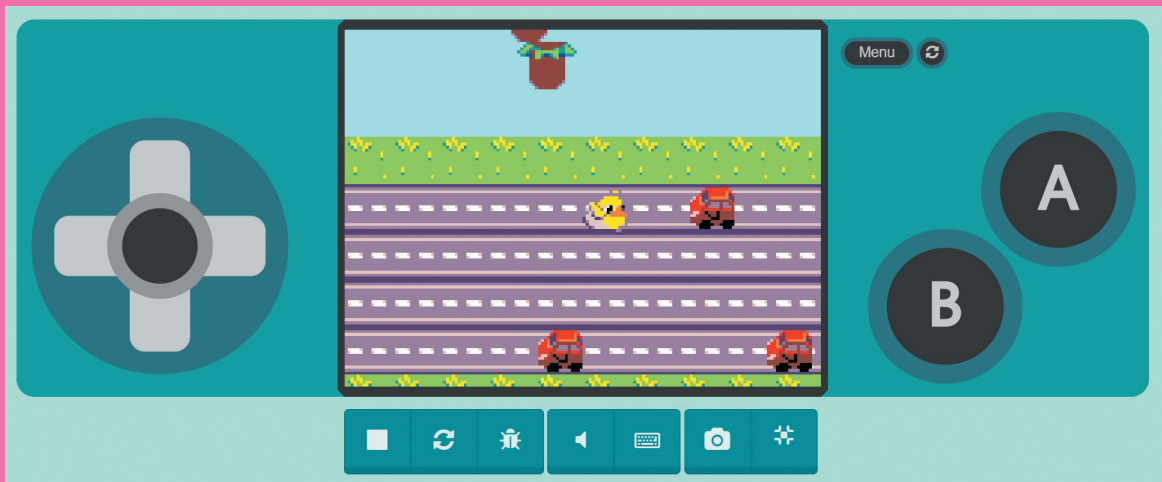
### 練習一下

請在「當遊戲更新 1000ms」事件中加入另外一種不同的木頭髮射物（共二種），並隨機出現其中一種木頭髮射物。

# 6 Chapter

## 「小鴨過街」遊戲

本章學習重點在於使用較大背景圖的直軸遊戲；使用按鈕事件移動角色；運用奇偶數運算配合判斷式在河流中指定木頭的移動方向等，這些都是設計遊戲常用的技巧。



## 遊戲說明

天快黑了，小鴨要趕緊回去溫暖的家，小鴨現在的位置在地圖最下方，小鴨的家在地圖的最左上方，小鴨回家時要經過大馬路，不能被車撞到；也必須要游過小河，小河中有木頭也不能被木頭撞到，過了河到達灌木（Surue）才算安全到家。

## 6.1 遊戲設計分析

### 一、遊戲角色：

角色名稱	角色類別	素材	創建角色的事件
小鴨	Player		當啟動時，初始位置（88,248）
小鴨家	Home		當啟動時，初始位置（8,8）
汽車	Enemy		當遊戲更新間隔 500ms
木頭	Enemy		當遊戲更新間隔 1000ms

### 二、遊戲事件：

小鴨過街

當啟動時(產生遊戲主角小鴨、小鴨家、繪製背景。)

當遊戲更新間隔 500ms(產生汽車角色，隨機出現在任一車道上，並由右方出現。)

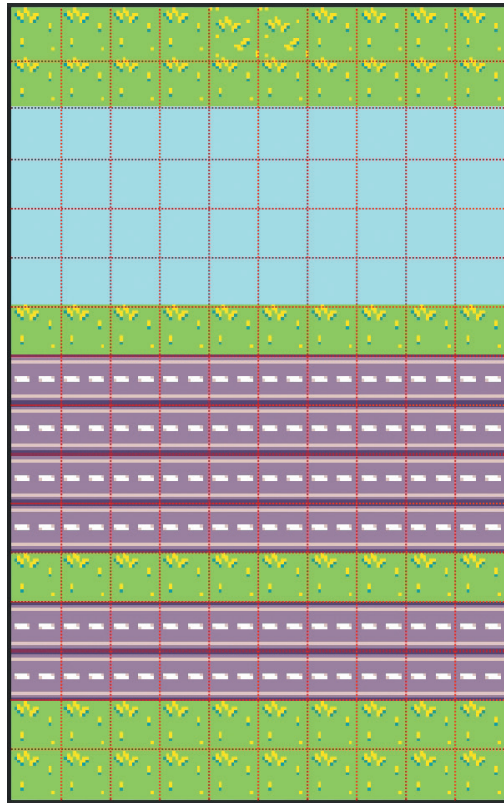
當遊戲更新間隔 1000ms(產生木頭角色，隨機出現在任一水道上，並依不同水道而有不同的移動方向。)

角色類別重疊(偵測小鴨和汽車、木頭是否碰撞，如果碰撞顯示遊戲結果為「輸」。)

角色類別重疊(偵測小鴨和灌木(Shrub)是否碰撞，如果碰撞顯示遊戲結果為「贏」。)

### 三、背景設計：

貼圖地圖為  $10 \times 16$  塊（直長條形）地圖，底圖為草地貼圖，下方放置車道，上方放置河流貼圖，如下圖所示：



## 6.2 「當啟動時」事件

遊戲一開始時屏幕顯示背景、創建遊戲角色及設定角色物理動作等。

### 程式碼說明

#### ▼行號 01

當啟動時執行事件區塊內的程式

當啟動時



#### ▼行號 03

創建遊戲主角「小鴨」，角色類別預設為「Player」

#### ▼行號 04

設定「小鴨」的初始位置在地圖的最下方

#### ▼行號 05

設定「小鴨」只能在屏幕中移動，避免主角「小鴨」離開屏幕

#### ▼行號 06

鏡頭跟隨角色，讓視角隨「小鴨」在地圖中移動

#### ▼行號 07

創建「家」的貼圖，讓「小鴨」可以回家

#### ▼行號 08

設定「家」的貼圖位置在地圖的最左上方，由於貼圖的解析度為  $16 \times 16$  像素，所以「家」貼圖中心點的座標設為 (8,8)

#### ▼行號 02

繪製遊戲背景

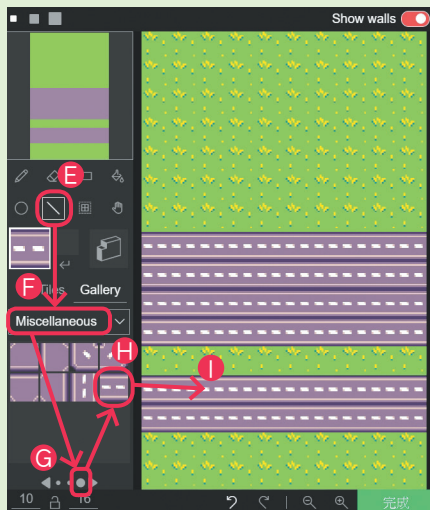
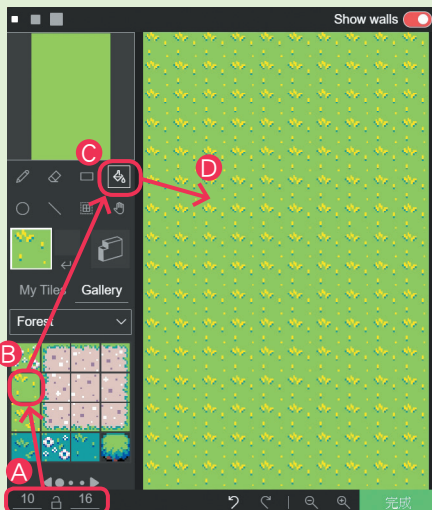
① 使用「場景」積木盒中的「貼圖地圖設為 ..」積木指令，點選空白貼圖開啟「地圖編輯器」

② 繪製草地

- A 設定地圖為「 $10 \times 16$ 」塊貼圖
- B 點選「草地」貼圖
- C 選擇「填滿」工具
- D 點選地圖填滿「草地」貼圖

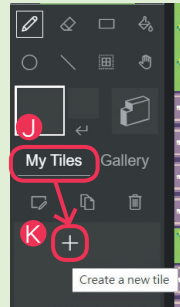
③ 繪製橫向馬路

- E 選擇「畫線」工具
- F 下拉選擇「Miscellaneous」圖庫
- G 移動至圖庫最後一頁
- H 選擇「橫向」馬路貼圖
- I 繪製六條橫向馬路



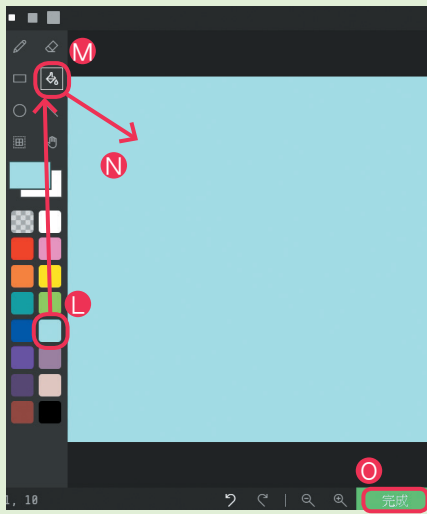
## ④ 選擇自訂圖庫

- J 選擇「My Tiles」（我的貼圖庫）
- K 選擇「+」（建立新貼圖），開啟貼圖編輯器



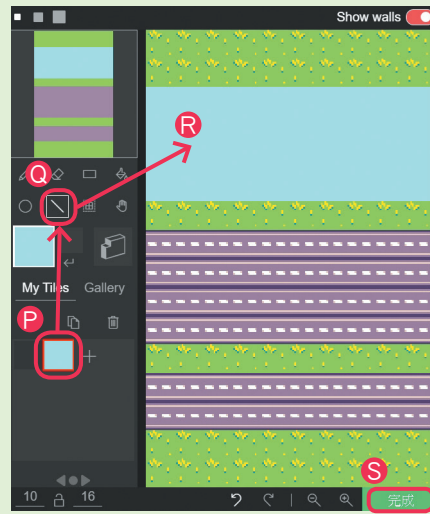
## ⑤ 繪製河流貼圖

- L 顏色選擇「Color 9」（水藍色）
- M 選擇「填滿」工具
- N 點選空白編輯區，填滿「水藍色」
- O 按「完成」回到地圖編輯器



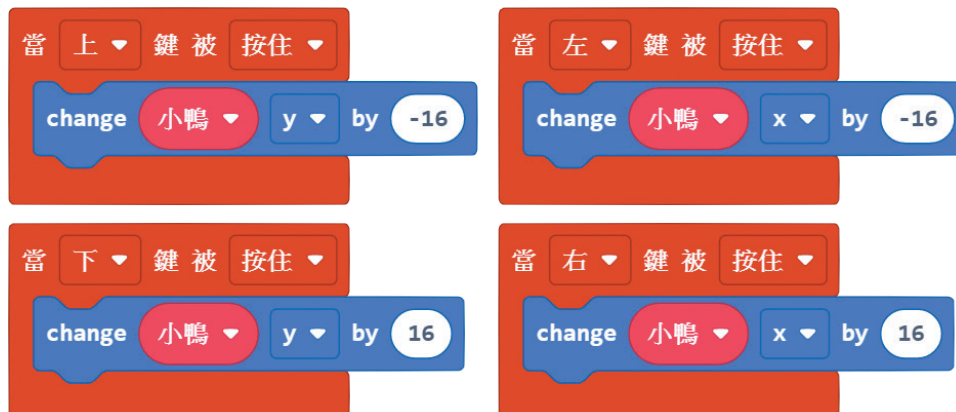
## ⑥ 繪製河流

- P 選擇「水藍色」貼圖
- Q 選擇「畫線」工具
- R 繪製四條橫向河流
- S 按「完成」回到程式編輯器



## 6.3 當方向按鈕「按下」事件

此遊戲不使用「控制器移動角色」積木指令，而是分別使用按鈕事件來控制主角「小鴨」的移動距離，這樣的做法可以讓「小鴨」移動時定位更精確，在貼圖地圖中每個貼圖解析度為  $16 \times 16$  像素，所以，使用「角色」積木盒中的「change 改變屬性」積木指令，每次按方向鍵時固定讓「小鴨」從原位置移動 16 像素到達與方向鍵對應的新位置。



## 6.4 「當遊戲更新 500 毫秒」事件

每隔 500 毫秒（0.5 秒）讓車子隨機出現在屏幕中的任一「馬路」貼圖上，車子開始的位置設定在屏幕的右側，並設定車子往左移動。

### 程式碼說明

#### ▼行號 01

當每隔 0.5 秒時執行事件區塊內的程式

#### ▼行號 02

創建「車子」角色，並設定角色類別為「Enemy」（敵人）

#### ▼行號 03

設定「車子」角色移動至屏幕側邊時自動刪除

#### ▼行號 04

設定剛創建的「車子」角色隨機放置在地圖中的任一「馬路」貼圖上面

#### ▼行號 05

設定「車子」角色移動速度為往左移動

#### ▼行號 06

設定「車子」角色從屏幕最右側出現



## 6.5 「當遊戲更新 1000 毫秒」事件

每隔 1 秒讓木頭隨機出現在屏幕中任一「河流」貼圖的左側或右側（以隨機產生的數字是偶數或奇數區分），並設定木頭往對向移動。

### 程式碼說明

#### ▼行號 02

新增一個名為「列」的變數，變數內容設為隨機取 2 到 5 之間的數，用於設定「木頭」貼圖的垂直位置列

#### ▼行號 03

創建「木頭」角色，並設定角色類別為「Enemy」（敵人）

#### ▼行號 04

設定「木頭」角色移動至屏幕側邊時自動刪除

#### ▼行號 05

判斷木頭要從哪邊出現，「列」變數內容為偶數時，「木頭」角色從右側出現；「列」變數內容為奇數時，「木頭」角色從左側出現（偶數條件設定：「列」變數內容  $\div 2$  的餘數等於 0）

#### ▼行號 06

當條件成立時（「列」變數內容為偶數），設定「木頭」角色移動速度為「-30」（往左移動）

#### ▼行號 07

並且設定「木頭」角色貼圖的左邊緣距屏幕左側 160 像素（從最右側出現）

#### ▼行號 08

當條件不成立時（「列」變數內容不為偶數），設定「木頭」角色移動速度為「30」（往右移動）

#### ▼行號 01

當每隔 1000 毫秒（1 秒）時執行事件區塊內的程式

```

當遊戲更新 間隔 1000 毫秒
  變數 列 設為 隨機取數 2 到 5
  變數 木頭 設為 角色 類別 Enemy
  角色 木頭 標記 撞牆時摧毀 設為 開
  如果 列 ÷ 2 的餘數 = 0 那麼
    set 木頭 速度 x (vx) to -30
    set 木頭 左邊 to 160
  否則
    set 木頭 速度 x (vx) to 30
    set 木頭 右邊 to 0
  set 木頭 上 to 16 x 列
  
```

#### ▼行號 09

並且設定「木頭」角色貼圖的右邊緣距屏幕左側 0 像素（從最左側出現）

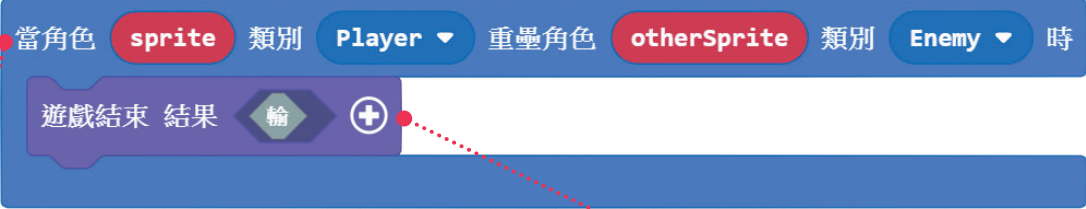
#### ▼行號 10

離開判斷式後，設定「木頭」角色貼圖的垂直位置，貼圖上緣距離屏幕上方的位置為 16 乘於「列」變數內容（每個貼圖解析度為  $16 \times 16$  像素），主要作用是將「木頭」角色貼圖往下移至地圖的河流貼圖中

## 6.6 「當類別 Player 重疊類別 Enemy」事件

當小鴨被車子或木頭撞到時，顯示「GAME OVER」遊戲結束。

### 程式碼說明



The code block is a blue Scratch event block with the following structure: "當角色" (When a character) - "sprite" (sprite) - "類別" (category) - "Player" (dropdown) - "重疊角色" (collides with) - "otherSprite" (otherSprite) - "類別" (category) - "Enemy" (dropdown) - "時" (when). Below this is a purple "遊戲結束 結果" (Game Over Result) block with a dropdown set to "輸" (Loss) and a plus sign icon. Two red dotted lines connect the "Player" and "Enemy" dropdowns to their respective explanation boxes below.

▼行號 01  
當玩家類別（Player，小鴨）碰到敵人類別（Enemy，車子、木頭）時執行事件區塊內的程式

▼行號 02  
屏幕顯示「GAME OVER」遊戲結束

## 6.7 「當類別 Player 重疊類別 Home」事件

當小鴨回到家時，顯示「YOU WIN」遊戲結束。

### 程式碼說明



The code block is a blue Scratch event block with the following structure: "當角色" (When a character) - "sprite" (sprite) - "類別" (category) - "Player" (dropdown) - "重疊角色" (collides with) - "otherSprite" (otherSprite) - "類別" (category) - "Home" (dropdown) - "時" (when). Below this is a purple "遊戲結束 結果" (Game Over Result) block with a dropdown set to "贏" (Win) and a plus sign icon. Two red dotted lines connect the "Player" and "Home" dropdowns to their respective explanation boxes below.

▼行號 01  
當玩家類別（Player，小鴨）碰到家類別（Home，家）時執行事件區塊內的程式

▼行號 02  
屏幕顯示「YOU WIN」遊戲結束

## 6.8 完整程式碼

〔程式檔：06 小鴨過街〕

The code is organized into several functional blocks:

- When the game starts (當啟動時):**
  - Set the background (貼圖地圖設為).
  - Set the variable 'Little Duck' (變數 小鴨) to a 'Player' character (角色 小鴨) with category 'Player' (類別 Player).
  - Set the position of 'Little Duck' (角色 小鴨) to x=88, y=248.
  - Set the 'Stay on screen' (標記 留在畫面) flag for 'Little Duck' to 'Always' (設定為).
  - Set the camera to follow 'Little Duck' (鏡頭 跟隨 角色 小鴨).
  - Set the variable 'Home' (變數 家) to a 'Home' character (角色 家) with category 'Home' (類別 Home).
  - Set the position of 'Home' (角色 家) to x=8, y=8.
- Control Logic (Control blocks):**
  - When the 'Up' key (上) is pressed, change 'Little Duck's' (小鴨) y-coordinate by -16.
  - When the 'Left' key (左) is pressed, change 'Little Duck's' (小鴨) x-coordinate by -16.
  - When the 'Down' key (下) is pressed, change 'Little Duck's' (小鴨) y-coordinate by 16.
  - When the 'Right' key (右) is pressed, change 'Little Duck's' (小鴨) x-coordinate by 16.
- Collision Detection (When sprite touches other sprite):**
  - When a 'Player' sprite (sprite 類別 Player) touches an 'Enemy' sprite (otherSprite 類別 Enemy), the game ends with the result 'Lose' (遊戲結束 結果 輸).
  - When a 'Player' sprite (sprite 類別 Player) touches a 'Home' sprite (otherSprite 類別 Home), the game ends with the result 'Win' (遊戲結束 結果 贏).

程式碼續接下頁



```
當遊戲更新 間隔 500 毫秒
  變數 車子 設為 角色 類別 Enemy
  角色 車子 標記 撞牆時摧毀 設為 關
  角色 車子 放置到任意貼圖 上面
  set 車子 速度 x (vx) to -50
  set 車子 左邊 to 160

當遊戲更新 間隔 1000 毫秒
  變數 列 設為 隨機取數 2 到 5
  變數 木頭 設為 角色 類別 Enemy
  角色 木頭 標記 撞牆時摧毀 設為 關
  如果 列 ÷ 2 的餘數 = 0 那麼
    set 木頭 速度 x (vx) to -30
    set 木頭 左邊 to 160
  否則
    set 木頭 速度 x (vx) to 30
    set 木頭 右邊 to 0
  set 木頭 上 to 16 × 列
```

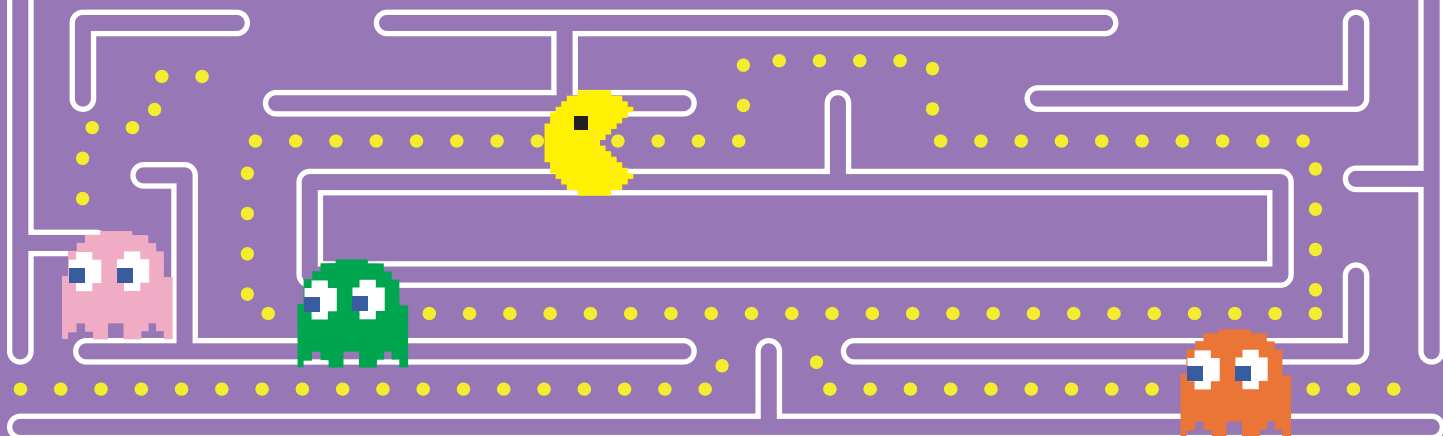
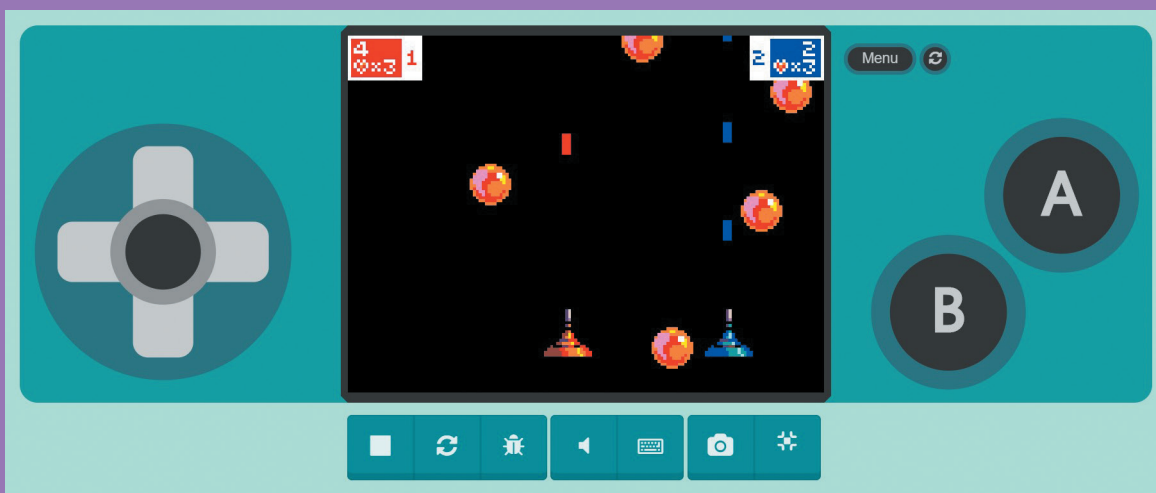
### 練習一下

請加入倒數計時 10 秒，10 秒內如果還沒讓小鴨回家，屏幕顯示「GAME OVER」遊戲結束。



## 「射擊雙打」遊戲

本章學習重點為多玩家積木（目前支援雙人連線）的使用；判斷兩個玩家任一方生命值为 0 則遊戲結束；透過角色的 z（深度）屬性參數來達到判斷哪個玩家得分？



## 遊戲說明

由二個玩家所組成的太空船艦隊，突襲了敵人的星球，敵人派出飛碟準備撞擊太空船保護自己的星球，太空船可以閃躲或是發射子彈將飛碟擊毀得 1 分，一旦玩家太空船被飛碟撞擊三次後遊戲結束。

## 7.1 遊戲設計分析

### 一、遊戲角色：

角色名稱	角色類別	素 材	創建角色的事件
玩家 1	Player		當啟動時，初始位置 (25,100)
玩家 2	Player		當啟動時，初始位置 (128,100)
飛碟	Enemy		當遊戲更新 間隔 500 毫秒
子彈 1	Projectile		當玩家 1 按下 A 鍵時
子彈 2	Projectile		當玩家 2 按下 A 鍵時

### 二、遊戲事件：

#### 射擊雙打

當啟動時(產生二個遊戲玩家角色、設定生命值。)

當玩家 1 的 A 鍵被按住(玩家 1 往上發射子彈，並設定子彈深度為 1，做為玩家 1 得分依據。)

當玩家 2 的 A 鍵被按住(玩家 2 往上發射子彈，並設定子彈深度為 2，做為玩家 2 得分依據。)

當遊戲更新間隔 500 毫秒(產生飛碟發射物，隨機出現在屏幕上方，並隨機更改飛碟速度。)

當玩家類別重疊飛碟類別(偵測玩家和飛碟是否碰撞，如果碰撞該玩家生命值 -1，該飛碟刪除。)

當子彈類別重疊飛碟類別(偵測子彈和飛碟是否碰撞，如果碰撞該玩家得分 +1，該子彈和飛碟刪除。)

## 7.2 「當啟動時」事件

遊戲一開始時創建遊戲角色及設定角色物理動作等，並設定玩家生命值。

### 程式碼說明

#### ▼行號 03

創建遊戲主角「玩家 2」，至「素材庫」選擇「spaceGreenShip」（綠色太空船）貼圖

#### ▼行號 04

使用「控制器」積木盒中多個玩家的「玩家.. 控制器移動角色」積木指令，下拉角色名稱選擇「玩家 1」，讓控制器與角色「玩家 1」做連結，能使用方向鍵移動「玩家 1」太空船在屏幕中的位置

#### ▼行號 05

讓控制器與角色「玩家 2」做連結，此時，在模擬器上移動二艘太空船的控制方式可使用如下鍵盤：

玩家 1		玩家 2	
↑	W 上	I	上
↓	S 下	K	下
→	A 左	J	左
←	D 右	L	右

#### ▼行號 06

將「玩家 1」生命值設為 3

#### ▼行號 07

將「玩家 2」生命值設為 3

#### ▼行號 08

將「玩家 1」設置在座標（25,100）位置

#### ▼行號 09

將「玩家 2」設置在座標（128,100）位置

#### ▼行號 01

當啟動時執行事件區塊內的程式

#### ▼行號 02

創建遊戲主角「玩家 1」，至「素材庫」選擇「spaceOrangeShip」（橘色太空船）貼圖

當啟動時

- 變數 玩家1 設為 角色 (spaceOrangeShip) 類別 Player
- 變數 玩家2 設為 角色 (spaceGreenShip) 類別 Player
- 玩家 1 控制器移動角色 玩家1
- 玩家 2 控制器移動角色 玩家2
- 玩家 1 的生命設為 3
- 玩家 2 的生命設為 3
- 角色 玩家1 的位置設為 x 25 y 100
- 角色 玩家2 的位置設為 x 128 y 100
- 角色 玩家1 標記 留在畫面 設為 開
- 角色 玩家2 標記 留在畫面 設為 開

#### ▼行號 10

設定「玩家 1」不能移出屏幕外

#### ▼行號 11

設定「玩家 2」不能移出屏幕外

## 7.3 「當玩家 1 按下 A 鍵時」事件

當「玩家 1」按下「A」鍵時，發出子彈（發射物）並模擬發射子彈的聲音，為讓重疊事件發生時能知道是被誰的子彈擊中，將「玩家 1」的發射子彈「z（深度）」設定為「1」。

### 程式碼說明

#### ▼ 行號 01

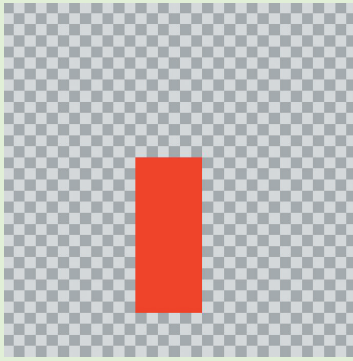
使用「控制器」積木盒多個玩家中的「當玩家按下 A 鍵時」事件積木，設定當「玩家 1」按下「A」鍵時執行事件區塊內的程式

#### ▼ 行號 02

使用「音效」積木盒中的「撥放音效」積木指令，撥放「pew pew」音效

#### ▼ 行號 03

創建「子彈 1」發射物來自「玩家 1」，並設定往上垂直速度（vy）  
點選發射物空白貼圖，開啟貼圖編輯器，使用畫筆工具畫出如下子彈：



#### ▼ 行號 04

深度 z 為距離鏡頭的距離，數值愈大，貼圖在愈上面，深度在這裡是做為重疊事件發生時，判斷是哪個玩家得分



## 7.4 「當玩家 2 按下 A 鍵時」事件

當「玩家 2」按下「A」鍵時，發出子彈（發射物）並模擬發射子彈的聲音，為讓重疊事件發生時能知道是被誰的子彈擊中，將「玩家 2」的發射子彈「z（深度）」設定為「2」。

### 程式碼說明

#### ▼行號 01

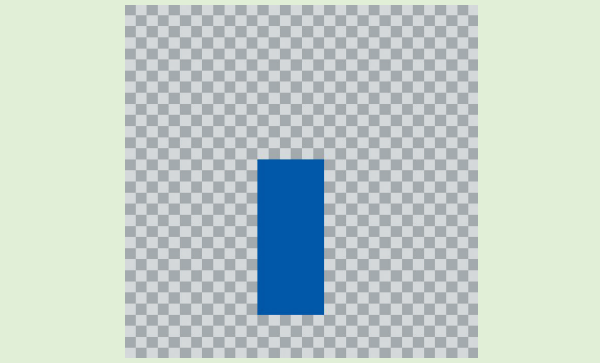
使用「控制器」積木盒多個玩家中的「當玩家按下 A 鍵時」事件積木，設定當「玩家 2」按下「A」鍵時執行事件區塊內的程式

#### ▼行號 02

使用「音效」積木盒中的「播放音效」積木指令，播放「pew pew」音效

#### ▼行號 03

創建「子彈 2」發射物來自角色，並設定往上垂直速度（vy）  
點選發射物空白貼圖，開啟貼圖編輯器，使用畫筆工具畫出如下子彈：



#### ▼行號 04

深度 z 為距離鏡頭的距離，數值愈大，貼圖在愈上面，深度在這裡是做為重疊事件發生時，判斷是哪個玩家得分



當有第二個玩家時，在模擬器上 A、B 鍵的控制方式可使用如下鍵盤：



## 7.5 「當遊戲更新間隔 500 毫秒」事件

遊戲執行中，每隔 0.5 秒屏幕上方產生一個飛碟（發射物）往下飛出，發射物預設類別均為「Projectile」，為與同是發射物的子彈區分，以方便後面設計玩家與飛碟、子彈與飛碟的重疊事件，更改飛碟角色類別為「Enemy」。

### 程式碼說明

#### ▼ 行號 01

當遊戲運行中每隔 0.5 秒執行事件區塊內的程式

#### ▼ 行號 02

創建「飛碟」發射物來自側邊，並設定「飛碟」隨機垂直往下速度



#### ▼ 行號 03

設定「飛碟」產生時是在屏幕的寬度範圍內，水平位置 (x) 設定為隨機取「10」到「150」的數值

#### ▼ 行號 04

為與子彈角色類別區分，更改飛碟角色類別為「Enemy」

## 7.6 「當玩家類別重疊飛碟類別」事件

當某一玩家太空船與飛碟發生碰撞時，該玩家生命值 -1，並且將與玩家重疊的飛碟角色刪除，刪除飛碟角色要使用類別為「Enemy」前面的角色代稱（otherSprite），不能直接使用角色名稱，否則屏幕上相同名稱的角色都會被刪除，最後判斷二個玩家生命值是否為 0，如果任何一個玩家生命值为 0，則遊戲結束。

### 程式碼說明

#### ▼行號 01

當玩家類別 (Player) 重疊飛碟類別 (Enemy) 時執行事件區塊內的程式

#### ▼行號 02

模擬太空船撞到飛碟時的聲音

#### ▼行號 03

為避免重複將玩家生命值 -1，需立即將與玩家重疊的飛碟角色刪除

#### ▼行號 04

判斷是否為「玩家 1」與飛碟發生碰撞，如果條件成立則執行「那麼」區塊內的程式

#### ▼行號 05

將「玩家 1」生命值 -1



#### ▼行號 06

如果條件不成立則執行「否則」區塊內的程式 (玩家 2 與飛碟發生碰撞)

#### ▼行號 07

將「玩家 2」生命值 -1

#### ▼行號 08

產生畫面震動特效

#### ▼行號 09

判斷二個玩家的生命值，如果「玩家 1 生命值为 0 或玩家 2 生命值为 0」，則執行區塊內的程式

#### ▼行號 10

遊戲結束，按「+」設定結束時的效果為「暴雪 (blizzard)」

## 7.7 「當子彈類別重疊飛碟類別」事件

當任一玩家射出的子彈打中飛碟時，則該玩家得分 +1，並將該子彈與飛碟都刪除，這個事件使用子彈貼圖的深度（z）屬性的來判斷子彈是由哪個玩家發射的。

### 程式碼說明

#### ▼行號 02

模擬飛碟被子彈擊中的聲音

#### ▼行號 03

將產生碰撞的子彈刪除，避免繼續往上飛造成一顆子彈射中多個飛碟（將「角色類別重疊」事件積木中的角色代稱「sprite」拖曳至「角色摧毀」積木指令參數中）

#### ▼行號 04

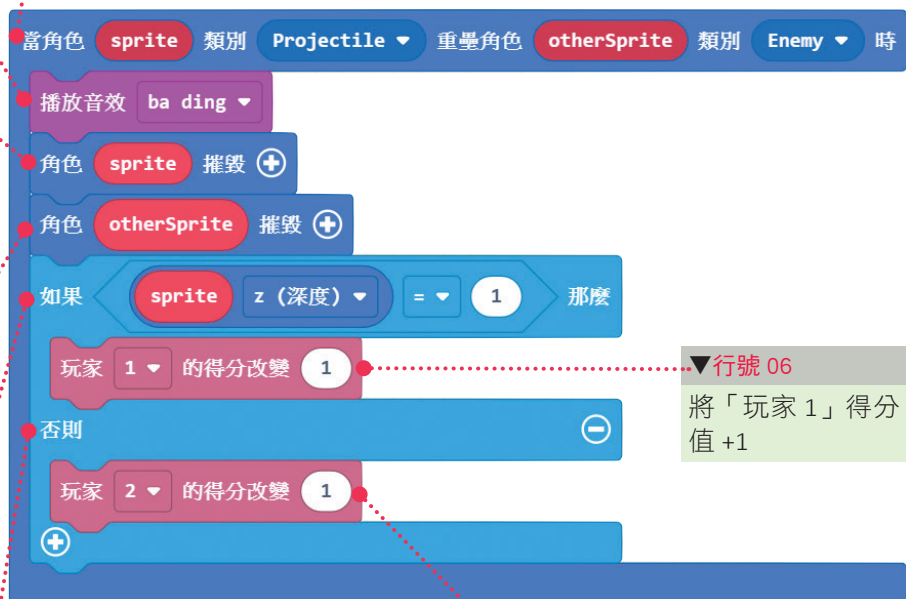
將產生碰撞的飛碟刪除（將「角色類別重疊」事件積木中的角色代稱「otherSprite」拖曳至「角色摧毀」積木指令參數中）

#### ▼行號 05

判斷產生碰撞的子彈是否為「玩家 1」射出（深度 z 為 1），如果條件成立則執行「那麼」區塊內的程式

#### ▼行號 01

當子彈類別（Projectile）重疊飛碟類別（Enemy）時執行事件區塊內的程式



```

當角色 sprite 類別 Projectile 重疊角色 otherSprite 類別 Enemy 時
  播放音效 ba ding
  角色 sprite 摧毀 +
  角色 otherSprite 摧毀 +
  如果 sprite z (深度) = 1 那麼
    玩家 1 的得分改變 1
  否則
    玩家 2 的得分改變 1
  
```

#### ▼行號 06

將「玩家 1」得分值 +1

#### ▼行號 07

如果條件不成立則執行「否則」區塊內的程式（飛碟是被玩家 2 的子彈擊中）

#### ▼行號 08

將「玩家 2」得分值 +1

## 7.8 完整程式碼

〔程式檔：07 射擊雙打〕

The image displays a series of Scratch code blocks for a shooting game. The code is organized into three main sections:

- Initialization (When Game Starts):** This section sets up the game environment. It includes:
  - Two variables, 'Player 1' and 'Player 2', both set to a 'Player' character.
  - Two 'Player 1' and 'Player 2' controller blocks to manage character movement.
  - Two 'Set Life' blocks, each setting the life of a player to 3.
  - Two 'Set Position' blocks: Player 1 is positioned at x=25, y=100; Player 2 is positioned at x=128, y=100.
  - Two 'Set Mark' blocks, each setting the 'Mark' property to 'Stay on Screen'.
- Player 1 Shooting (When A Key is Pressed):** This block triggers the following actions:
  - Play the 'pew pew' sound effect.
  - Create a 'Bullet 1' projectile from Player 1, with vx=0 and vy=-200.
  - Set the 'z (depth)' of 'Bullet 1' to 1.
- Player 2 Shooting (When A Key is Pressed):** This block triggers the following actions:
  - Play the 'pew pew' sound effect.
  - Create a 'Bullet 2' projectile from Player 2, with vx=0 and vy=-200.
  - Set the 'z (depth)' of 'Bullet 2' to 2.
- Game Update (Every 500 ms):** This block handles periodic updates:
  - Create a 'Disc' projectile from the side, with vx=0 and vy set to a random value between 10 and 100.
  - Set the 'x' position of 'Disc' to a random value between 10 and 150.
  - Set the 'Category' of 'Disc' to 'Enemy'.

程式碼續接下頁



```
當角色 sprite 類別 Projectile 重疊角色 otherSprite 類別 Enemy 時
  播放音效 ba ding
  角色 sprite 摧毀
  角色 otherSprite 摧毀
  如果 sprite z (深度) = 1 那麼
    玩家 1 的得分改變 1
  否則
    玩家 2 的得分改變 1
```



```
當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Enemy 時
  播放音效 pew pew
  角色 otherSprite 摧毀
  如果 sprite = 玩家1 那麼
    玩家 1 的生命改變 -1
  否則
    玩家 2 的生命改變 -1
  鏡頭 震動幅度 4 像素 持續 500 毫秒
  如果 玩家 1 的生命 = 0 或 玩家 2 的生命 = 0 那麼
    遊戲結束 結果 輸 效果 暴雪 (blizzard)
```

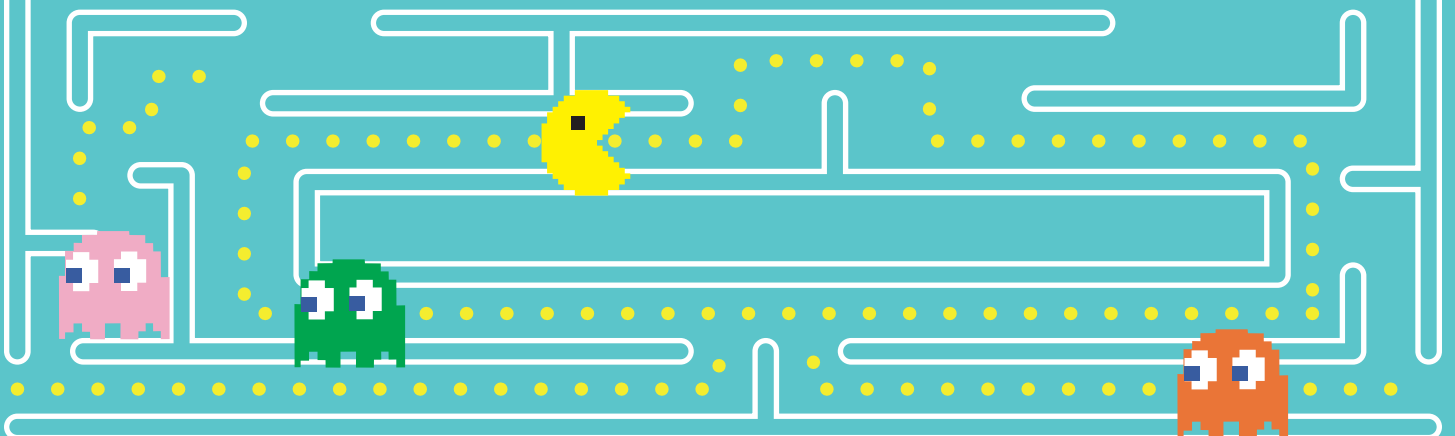
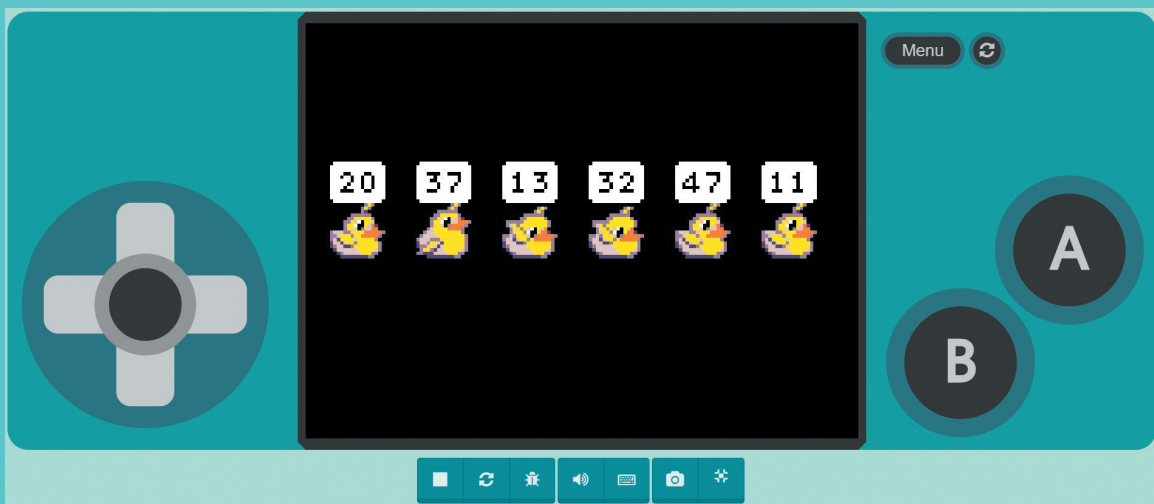
## 練習一下

請加入當飛碟產生時，飛碟會向左下和右下射出兩顆砲彈，任一玩家太空船如果被擊中生命值 -1。



## 「大樂透開獎」遊戲

本章學習重點在於如何使用迴圈將陣列變數中的圖形項目整齊排放至屏幕上；如何產生不重複之亂數值；使用「動畫」積木盒讓屏幕上的角色動起來。



## 遊戲說明

大樂透的基本玩法必須從 1 ~ 49 號碼中任選不重複的 6 個號碼進行投注，彩券公司將隨機開出 6 個號碼加 1 個特別號，本遊戲將模擬開獎單位隨機開出 6 個不重複的號碼。

## 8.1 遊戲設計分析

### 一、遊戲角色：

角色名稱	角色類別	素 材	創建角色的事件
鴨子 (0 ~ 5)	Player		當啟動時

### 二、遊戲事件：

大樂透開獎

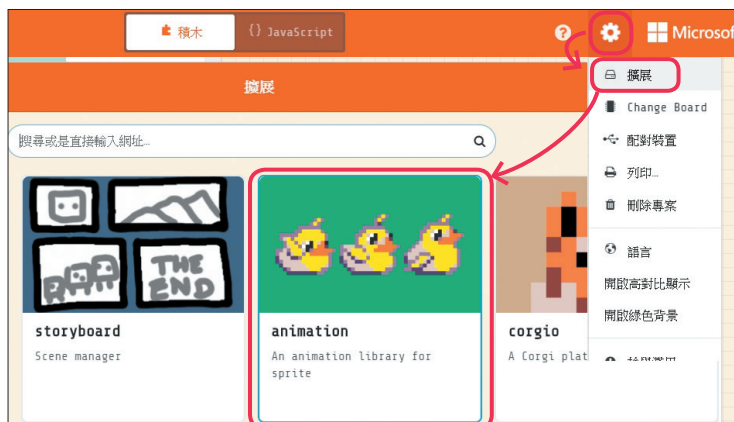
當啟動時(將 6 隻「鴨子」角色橫向固定於屏幕中間、鴨子拍翅膀畫面設定。)

當 A 鍵被按住時(將 6 個號碼設為 0，隨機產生不重複的 6 個號碼，分別讓 6 隻鴨子說出號碼，並播放拍翅膀動畫。)

### 三、擴展動畫積木盒

本章動畫功能需使用「動畫」積木盒，操作方式如下：

1. 「新增專案」後，點選程式編輯區右上角「齒輪」圖示，下拉選單選擇「擴展」。
2. 選擇「animation」（動畫）積木盒，積木區中將增加了一個「動畫」積木盒。



## 8.2 「當啟動時」事件

遊戲一開始創建遊戲角色 6 隻鴨子的陣列變數、動畫初始設定，屏幕上放置 6 隻鴨子。

### 程式碼說明

#### ▼行號 01

當啟動時執行事件區塊內的程式

#### ▼行號 02

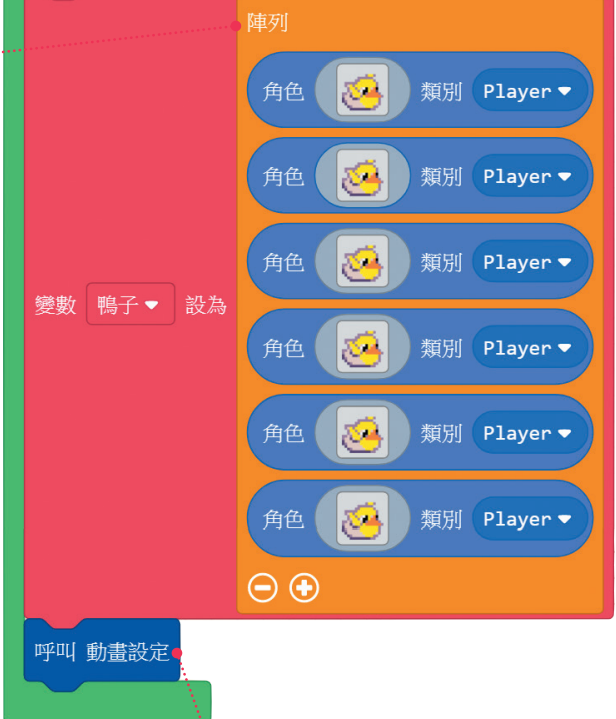
創建 6 個角色陣列

- ① 使用「陣列」積木盒中的「變數設為陣列」（數字或字串陣列都可以）
- ② 按「+」增加陣列項目到 6 個
- ③ 將變數名稱重新命名為「鴨子」
- ④ 暫時拖曳出「角色」積木盒中的「變數設為角色」積木指令，圖案使用素材庫中的「duck1」（鴨子 1）
- ⑤ 複製出 6 個角色積木，並分別拖曳至陣列項目中



- ⑥ 最後將「變數設為角色」積木指令刪除

#### 當啟動時



#### ▼行號 03

呼叫動畫設定函式，函式內將「鴨子」角色加入拍翅膀動畫和設定 6 隻鴨子的位置（呼叫函式前須先建立函式）

「動畫設定」自訂函式：

程式碼說明

▼行號 03  
新增第一張鴨子圖片的影格至「anim1」中，圖案使用素材庫中的「duck2」（鴨子2）

▼行號 04  
新增第二張鴨子圖片的影格至「anim1」中，圖案使用素材庫中的「duck3」（鴨子3）

▼行號 05  
新增第三張鴨子圖片的影格至「anim1」中，圖案使用素材庫中的「duck3」（鴨子4）

▼行號 06  
新增一個「Idle」（閒置）動作動畫

▼行號 07  
只新增一張鴨子圖片的影格至「anim2」中，使用素材庫中的「duck1」（鴨子1）

▼行號 08  
使用計次迴圈分別指定6隻鴨子角色，index變數會從0逐次增加1，6隻鴨子角色分別為鴨子(0)~鴨子(5)

▼行號 01  
當程式中呼叫自訂函式時執行函式區塊內的程式

▼行號 02  
建立一個「wings」（拍翅膀）動作的動畫，使用動畫前需先使用「擴展」方式新增「動畫」積木盒  
① 使用「動畫」積木盒中的「變數設為動畫」積木指令，新增一個「wings」動作動畫  
② 變數名稱重新命名為「anim1」，此為鴨子的wings（拍翅膀）動畫

▼行號 09  
利用 index 變數值指定鴨子角色，再將 anim1 動作的動畫附加到指定的鴨子角色中

▼行號 10  
利用 index 變數值指定鴨子角色，再將 anim2 動作的動畫附加到指定的鴨子角色中

▼行號 11  
將 6 隻鴨子橫向排列置於屏幕上，第 1 隻鴨子從第 15 像素開始放，每隻鴨子間隔 25 像素

## 8.3 「當按下 A 鍵時」事件

當按下「A」鍵時，產生 6 個不重複的號碼，並讓 6 隻鴨子輪流說出號碼，說出號碼時振動翅膀。

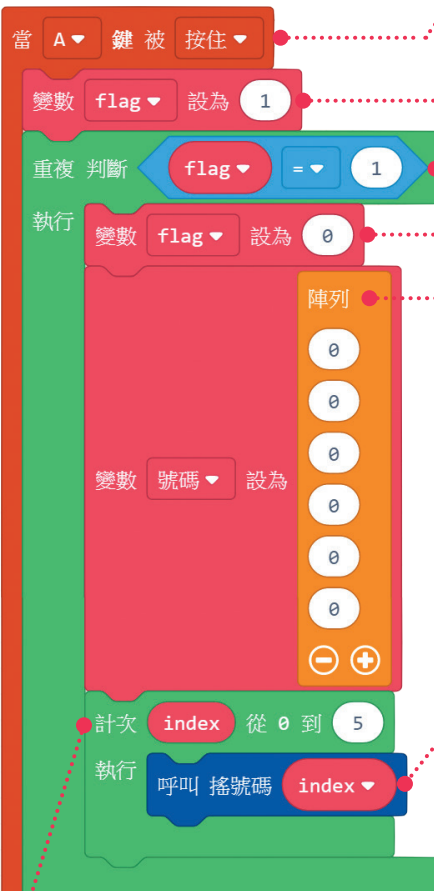
為了判斷後面產生的號碼不與前面重複，將新增一個「flag」變數做為辨識是否重複的標記，開始產生號碼前將 6 個號碼和「flag」變數設為 0。

每產生一個號碼就和之前的號碼做比對，如果號碼有重複，則先將「flag」變數設為 1，當產生完 6 個號碼後，再判斷「flag」是否為 1，如果是則重新產生 6 個號碼，如果「flag」為 0，則讓鴨子說出號碼。

### 程式碼說明

#### ▼行號 01

當按下「A」鍵時執行事件區塊內的程式



#### ▼行號 06

使用迴圈開始選號，共需 6 次 (0 ~ 5)

#### ▼行號 02

建立一個「flag」變數，做為判斷號碼是否重複的標記：1- 有重複，需重新選號；0- 沒有重複，完成選號

#### ▼行號 03

依條件判斷是否要繼續執行選號程式（如果 flag=1 則執行迴圈內的選號程式，否則離開迴圈）

#### ▼行號 04

選號前先預設重複的標記為號碼未重複 (flag=0)

#### ▼行號 05

建立一個陣列變數用來存放開獎號碼，號碼初始值均設為「0」

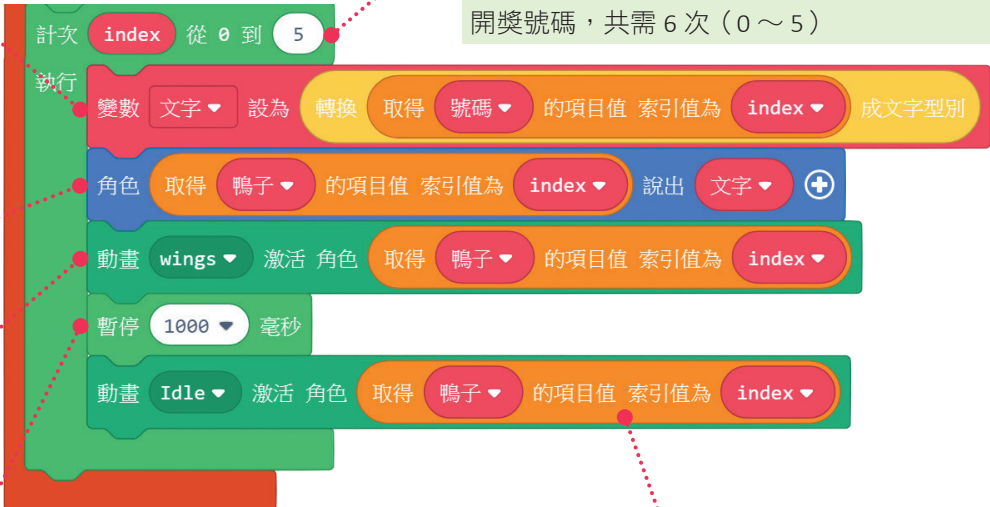
#### ▼行號 07

將選號程式寫在函式中，增加程式的閱讀性也更容易除錯

- ① 使用「函式」積木盒中的「建立一個函式」
- ② 建立名為「搖號碼」函式，並且添加一個數字 (Number) 參數，此參數告知函式目前欲產生的是第幾個號碼



- ③ 從「函式」積木盒拖曳出剛建立的函式呼叫指令
- ④ 拖曳「計次迴圈」積木指令中的「index」到函式的參數中



▼行號 08  
當程式離開重複判斷迴圈時，代表選號已經沒有重複，則準備讓 6 隻鴨子分別說出 6 個開獎號碼，共需 6 次 (0 ~ 5)

▼行號 09  
因為角色要說出的內容必須是文字，所以先取出要說出的號碼轉換為文字，並放在「文字」變數中

▼行號 10  
讓每隻鴨子說出對應的開獎號碼

▼行號 11  
讓正在說出號碼的鴨子，播放拍翅膀的動畫

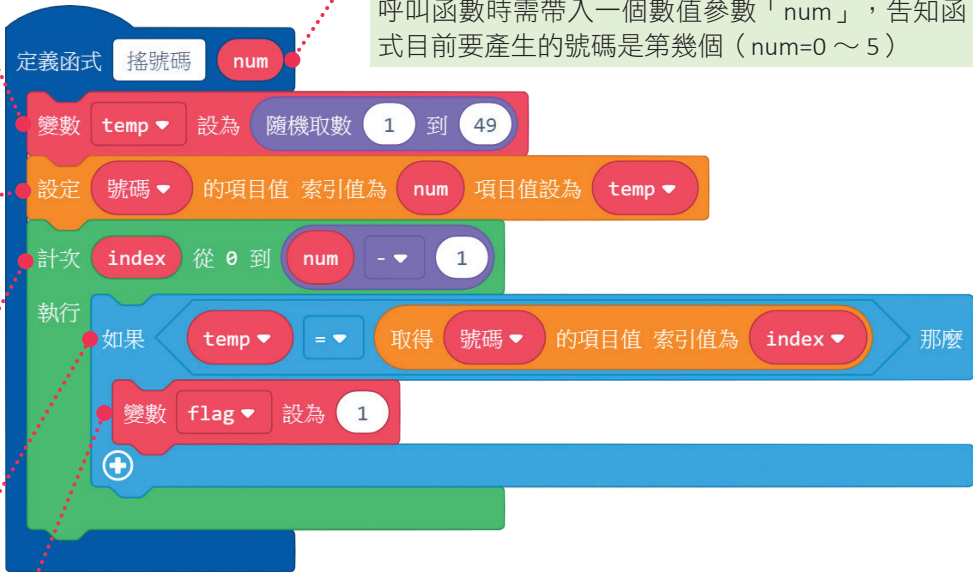
▼行號 12  
每隻鴨子說完號碼後暫停 1 秒再繼續下一隻鴨子，使用「迴圈」積木盒中的「暫停」積木指令，設定暫停「1000」毫秒 (1 秒)

▼行號 13  
讓鴨子停止拍翅膀

「搖號碼」自訂函式：

傳入的「num」參數為第幾個要產生的號碼，先暫時將隨機產生的號碼放到「temp」變數，再放到「號碼」陣列清單中指定 (num) 的位置，然後再一一檢查是否與前面的號碼重複。

### 程式碼說明



▼行號 01  
當程式中呼叫自訂函式時執行函式區塊內的程式，呼叫函數時需帶入一個數值參數「num」，告知函式目前要產生的號碼是第幾個 (num=0 ~ 5)

▼行號 02  
建立一個新的變數「temp」，此變數內容放置一個隨機號碼 (1 ~ 49)

▼行號 03  
將獲得的隨機號碼 (temp) 放到陣列第 num 個項目中

▼行號 04  
逐一判斷已開出的號碼中，是否曾經出現過「temp」變數內容

▼行號 05  
判斷隨機號碼 (temp) 是否和目前迴圈變數「index」所指定的號碼相等

▼行號 06  
如果隨機號碼 (temp) 和目前迴圈變數「index」所指定的號碼相等，則全部號碼要重新選號，因此將重複標記「flag」變數設為 1

## 8.4 完整程式碼

〔程式檔：08 大樂透開獎〕

```

當啟動時
    陣列
        角色 鴨子 類別 Player
        角色 鴨子 類別 Player
        角色 鴨子 類別 Player
        角色 鴨子 類別 Player
        角色 鴨子 類別 Player
        角色 鴨子 類別 Player
        角色 鴨子 類別 Player
    變數 鴨子 設為
    呼叫 動畫設定

定義函式 動畫設定
    變數 anim1 設為 動畫 wings 產生 間隔時間 200 毫秒
    添加影格 鴨子 到動畫 anim1
    添加影格 鴨子 到動畫 anim1
    添加影格 鴨子 到動畫 anim1
    變數 anim2 設為 動畫 Idle 產生 間隔時間 200 毫秒
    添加影格 鴨子 到動畫 anim2
    計次 index 從 0 到 5
    執行
        動畫 anim1 附加 角色 取得 鴨子 的項目值 索引值為 index
        動畫 anim2 附加 角色 取得 鴨子 的項目值 索引值為 index
        set 取得 鴨子 的項目值 索引值為 index x to index x 25 + 15
    
```

程式碼續接下頁

```

當 A 鍵被 按住
  變數 flag 設為 1
  重複判斷 flag = 1
  執行
    變數 flag 設為 0
    陣列
      0
      0
      0
      0
      0
      0
    變數 號碼 設為
    計次 index 從 0 到 5
    執行 呼叫 搖號碼 index
  計次 index 從 0 到 5
  執行
    變數 文字 設為 轉換 取得 號碼 的項目值 索引值為 index 成文字型別
    角色 取得 鴨子 的項目值 索引值為 index 說出 文字
    動畫 wings 激活 角色 取得 鴨子 的項目值 索引值為 index
    暫停 1000 毫秒
    動畫 Idle 激活 角色 取得 鴨子 的項目值 索引值為 index

定義函式 搖號碼 num
  變數 temp 設為 隨機取數 1 到 49
  設定 號碼 的項目值 索引值為 num 項目值設為 temp
  計次 index 從 0 到 num - 1
  執行
    如果 temp = 取得 號碼 的項目值 索引值為 index 那麼
      變數 flag 設為 1
  
```

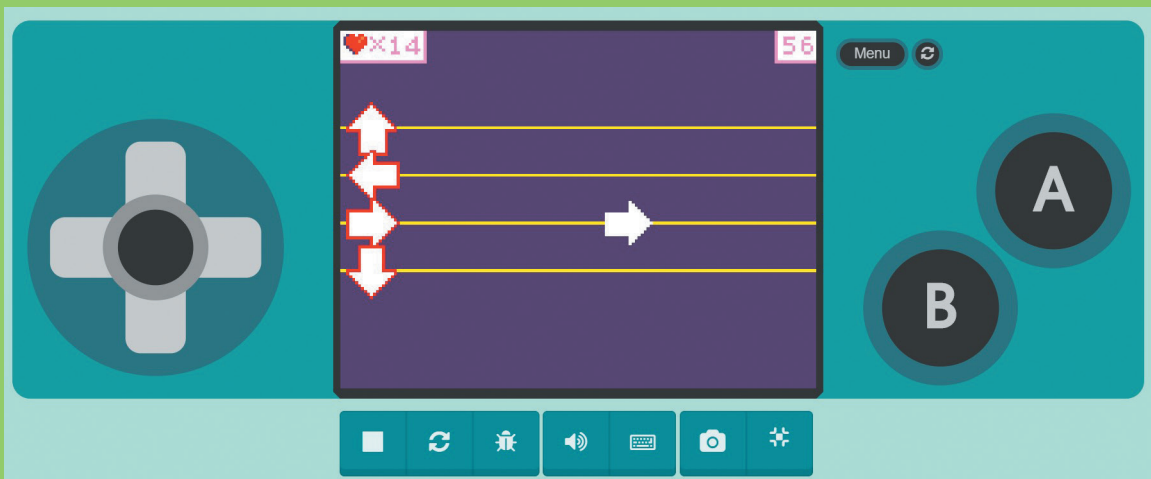
### 練習一下

請新增一個排序自訂函式，讓開獎號碼能由小到大排序。

# 9 Chapter

## 「指尖跳舞」遊戲

本章學習重點為如何更換陣列角色圖示；屏幕上繪製線條；自訂函數傳遞角色參數；多個角色重疊判斷等。





## 遊戲說明

市面上曾經流行的跳舞機是依螢幕上出現的箭頭搭配腳踩跳舞毯上的方向箭頭來配對得分，目前手機上也有很多相關 APP 遊戲，是依螢幕前方出現的箭頭，用觸控點擊對應的箭頭圖案來得分。

本章的遊戲在 Arcade 屏幕左側顯示上、下、左、右四種紅框箭頭圖案，當玩家按下方向鍵時，相對的箭頭將變為黃底箭頭圖案等待配對，在屏幕右側會隨機發出四種白色箭頭圖案往左移動，需在白色箭頭圖案消失在左側前使用方向鍵來配對得分。

## 9.1 遊戲設計分析

### 一、遊戲角色：

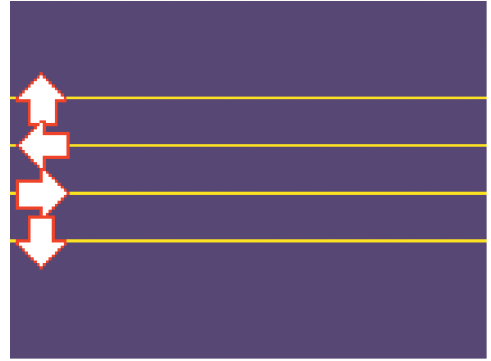
角色名稱	角色類別	素材	創建角色的事件
箭頭角色 (0 ~ 3)	Player		當啟動時
隨機箭頭 (0 ~ 3)	Projectile		重複無限次

### 二、遊戲事件：



### 三、背景設計：

遊戲背景色為暗紫色，屏幕中除了左側有四個紅框箭頭的角色貼圖，另外畫出四條黃色橫線，可讓遊戲更容易進行，完成後如右圖所示：



## 9.2 「當啟動時」事件

遊戲一開始屏幕顯示背景、創建遊戲角色、變數初始化及設定生命值等，最後顯示提示訊息等待按「A」鍵開始遊戲。

### 程式碼說明

#### ▼行號 01

當啟動時執行事件區塊內的程式

#### ▼行號 02

定義四個「箭頭角色」陣列及初始化三組箭頭圖案（自訂函式需先建立才能呼叫）

#### ▼行號 03

設定背景顏色、左側放置四個「箭頭角色」及繪製四條橫線（自訂函式需先建立才能呼叫）

#### ▼行號 04

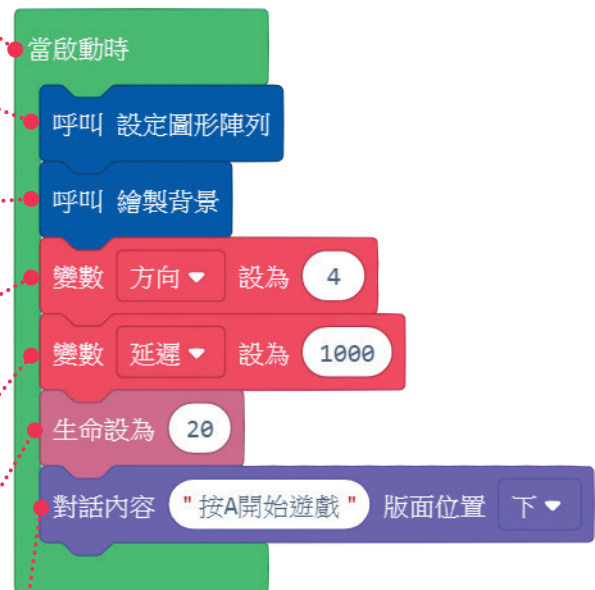
建立一個新的變數，內容代表使用者按下的方向鍵，分別如下：0- 上、1- 左、2- 右、3- 下、4- 未按，預設值為 4

#### ▼行號 05

建立一個新的變數，內容做為白色箭頭每隔多少時間從右側產生

#### ▼行號 06

生命設為「20」



#### ▼行號 07

在屏幕下方顯示提示訊息，等待玩家按「A」鍵開始遊戲


「設定圖形陣列」自訂函式：

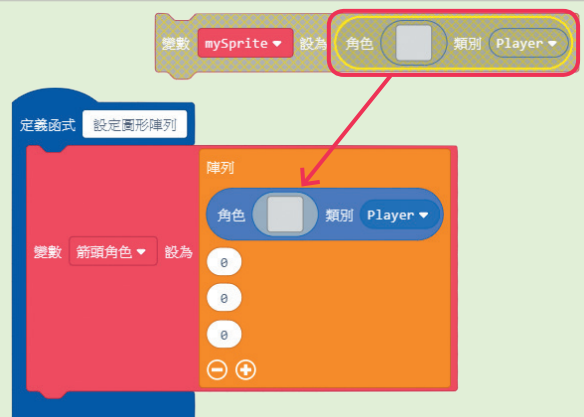
設定四個「箭頭角色」陣列變數和三組箭頭圖案陣列變數，三組箭頭圖案分別為：「白色箭頭」陣列圖案用於隨機產生的發射物圖像；「紅框箭頭」陣列圖案用於屏幕左側的角色圖像；「黃底箭頭」陣列變數用於按方向鍵時取代對應之「紅框箭頭」圖像。

### 程式碼說明

#### ▼行號 02

遊戲中為了能做碰撞偵測，新增四個角色，置於屏幕左側，為了方便撰寫程式，四個角色使用相同名稱，分別為：箭頭角色 (0) - 上、箭頭角色 (1) - 左、箭頭角色 (2) - 右、箭頭角色 (3) - 下，先不設定角色圖像：

- ① 使用「陣列」積木盒中的「變數設為..」積木指令（數字陣列或文字陣列都可以）
- ② 變數名稱更改為「箭頭角色」，按「+」新增陣列項目為四個
- ③ 先在程式編輯區中暫放一個「角色」積木盒中的「變數設為角色」積木指令
- ④ 再複製出四個  積木，並拖曳至陣列項目中



#### ▼行號 01

當程式中呼叫自訂函式時執行函式區塊內的程式

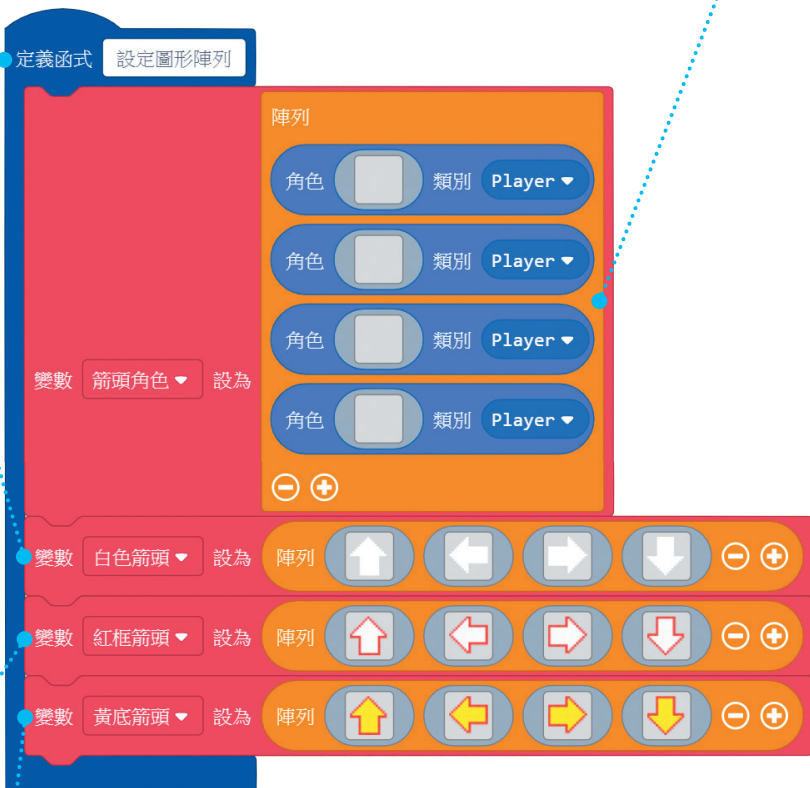
#### ▼行號 03

使用陣列變數，建立四個白色箭頭圖案，每個箭頭圖案解析度為 16×16 像素

- ① 使用「陣列」積木盒中的「變數設為..」積木指令（數字陣列或文字陣列都可以）
- ② 變數名稱更改為「白色箭頭」，按「+」新增陣列項目為四個
- ③ 項目內容使用「圖像」積木盒中的「空白」圖像積木
- ④ 點選空白圖像，開啟圖像編輯器分別繪製 4 個白色箭頭

#### ▼行號 04

複製上一個積木，開啟圖像編輯器分別將 4 個白色箭頭加入紅色外框，並改新變數名稱為「紅框箭頭」



#### ▼行號 05

複製上一個積木，開啟圖像編輯器分別將 4 個紅框箭頭內部填入黃色，並改新變數名稱為「黃底箭頭」

「繪製背景」自訂函式：

設定屏幕底色、屏幕左側放置四個箭頭角色（預設為紅框箭頭圖像）、屏幕上新增一塊畫布，在畫布上繪製四條黃色橫線，將四條黃色橫線的深度（z）置於四個箭頭角色之後。

### 程式碼說明

▼行號 01  
當程式中呼叫自訂函式時執行函式區塊內的程式

▼行號 02  
設定屏幕背景顏色為深紫色（#5c406c）

▼行號 03  
建立一個名為「畫布」的圖像，準備畫出四條黃色橫線（圖像必須設定給角色，才能將圖像顯示到屏幕上）

▼行號 04  
執行 4 次迴圈區塊內的程式，迴圈的預設變數 index 會從 0 逐次增加到 3

▼行號 05  
將四個箭頭角色分別放置在屏幕左側  
① 從「計次」迴圈積木指令拖曳 index 的變數名稱至索引值參數內  
② 組合出「(index\*16)+32」運算式，並放入垂直位置（y）參數中

▼行號 06  
建立箭頭角色時並未給定箭頭圖像，此時將分別設定紅框箭頭圖案給 4 個箭頭角色

▼行號 07  
在「畫布」的圖像中繪製 4 條黃色橫線，橫線位置分別間隔 16 像素

▼行號 08  
要將圖像顯示在屏幕上，必須將圖像設定到角色的貼圖中

▼行號 09  
由於程式是先放置 4 個紅框箭頭最後才繪製 4 條黃色線，所以 4 條黃色線會蓋在 4 個紅框箭頭上面，因此，將「線」角色的深度（z）屬性設為「-1」，讓 4 條黃線在紅框箭頭圖案的下層（角色預設的深度值為 0）

## 9.3 「重複無限次」事件

隨機產生四種白色箭頭發射物，並由屏幕右側移動至左側，依產生箭頭圖示放置於相對之橫線上。

### 程式碼說明

#### ▼行號 01

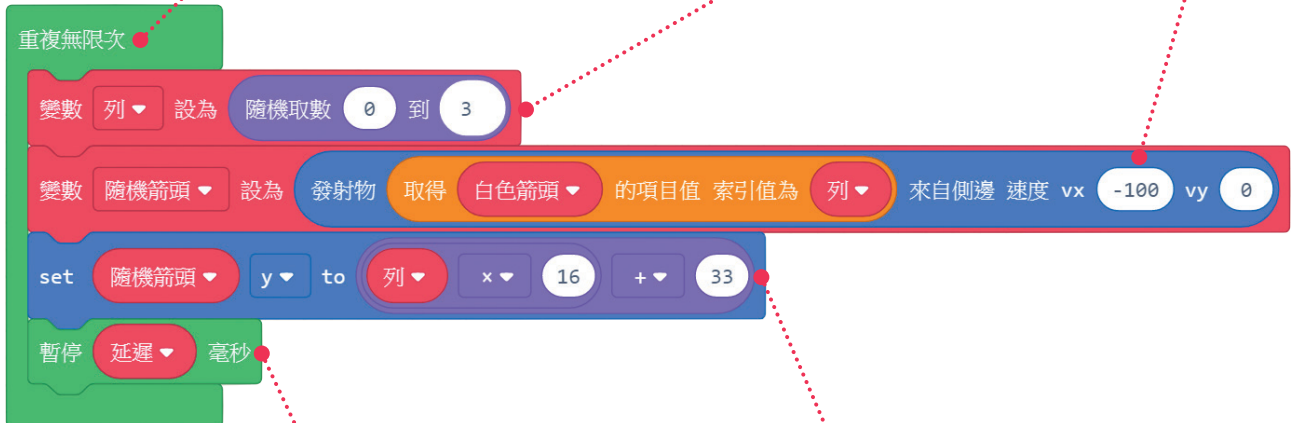
當程式執行時重複無限次執行事件區塊內的程式

#### ▼行號 02

新增一個「列」變數，變數內容隨機產生 0 ~ 3 數值，做為產生白色箭頭的方向，並且依照「列」變數內容，設定箭頭在屏幕的垂直位置

#### ▼行號 03

在屏幕右側產生隨機白色箭頭，往左側移動



The code block is a 'Repeat Forever' loop containing the following steps:

- Variable '列' (Column) is set to a random number between 0 and 3.
- A 'White Arrow' projectile is created from the right side with velocity vx = -100 and vy = 0.
- The 'y' position of the 'White Arrow' is set to '列 \* 16 + 33'.
- A 'Delay' block is used to pause the execution for a certain number of milliseconds.

#### ▼行號 05

白色箭頭產生的速度由「延遲」變數決定

#### ▼行號 04

設定白色箭頭的垂直位置，放置於對應的橫線上，組合出「(列 \* 16) + 33」運算式

## 9.4 「當方向鍵被按住時」事件

使用「函式」積木盒中的「建立一個函式」，修改函式名稱為「設定按鍵」函式，並新增一個數字（Number）參數。



當玩家按下方向鍵時，呼叫「設定按鍵」自訂函式，並帶入對應於該鍵之參數，此參數告知自訂函數玩家按了哪個方向鍵。



「設定按鍵」自訂函式：

當玩家按下方向鍵後先將屏幕左側四個箭頭角色均還原為紅框箭頭圖示，再將玩家所按下之方向鍵（`num` 參數內容）由紅框箭頭改為黃底箭頭，並設定「方向」變數為目前按鍵值，做為重疊角色判斷。

### 程式碼說明

#### ▼行號 01

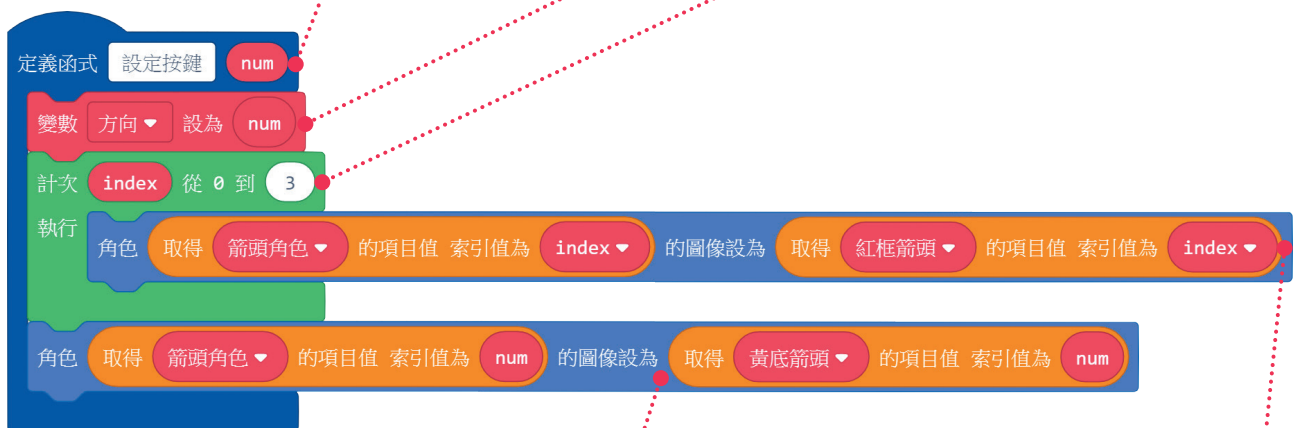
當程式中呼叫自訂函式時執行函式區塊內的程式，呼叫此函式必須帶入一參數，該參數會存入「`num`」變數中

#### ▼行號 02

設定「方向」變數內容為「`num`」變數內容，記錄玩家所按的方向鍵

#### ▼行號 03

共執行 4 次迴圈區塊內的程式，迴圈內的預設變數 `index` 會從 0 逐次增加到 3



#### ▼行號 05

依玩家所按下的方向鍵，將屏幕左側紅框箭頭角色的圖像，更改為對應的黃底箭頭圖像，索引值內容為「`num`」變數

#### ▼行號 04

由於玩家上一個按鍵會將對應的紅框箭頭改變為黃底箭頭，因此，在顯示下一個黃框箭頭前，先還原左側 4 個角色的圖像顯示為紅框箭頭。此積木可從「繪製背景」自訂函式中複製後修改

## 9.5 「當角色類別重疊時」事件

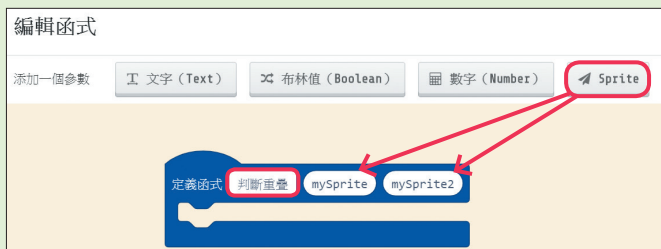
當屏幕左側的四個角色類別（`Player`）碰到白色箭頭發射物類別（`Projectile`）時，呼叫「判斷重疊」自訂函式，並帶入「玩家按下之角色」及發生碰撞之「發射物角色」二個角色參數，於「判斷重疊」自訂函式中判斷是否得分，如果玩家按的方向鍵與發生碰撞的白色箭頭發射物不同方向則生命值 -1。

## 程式碼說明

### ▼行號 03

使用自訂函式判斷兩個角色是否是同方向的箭頭，如果條件成立則得分 +1

- ① 使用「函式」積木盒中的「建立一個函式」，修改函式名稱為「判斷重疊」函式，並新增二個角色 (Sprite) 參數，先從「函式」積木盒中拖曳出剛建立的函式呼叫指令



- ② 依「方向」變數內容當作索引值，取出「箭頭角色」的角色名稱，放入第一個角色參數
- ③ 從重疊事件積木指令拖曳出「otherSprite」，將產生碰撞的白色箭頭放入第二個角色參數

### ▼行號 01

當「Player」類別重疊「Projectile」類別時執行事件區塊內的程式

### ▼行號 02

玩家如有按下任何方向鍵（方向 = 0 ~ 3），則執行條件成立區塊內程式

### ▼行號 04

當有重疊事件發生，但玩家未按任何方向鍵，則執行條件不成立區塊內程式

### ▼行號 05

讓生命值 -1

### ▼行號 06

撥放「pew pew」音效

### ▼行號 07

從重疊事件積木指令拖曳出「otherSprite」放入參數中，將發生重疊的白色箭頭發射物刪除，避免持續執行重疊事件

### ▼行號 08

將「方向」變數內容設定為「4」（玩家沒按方向鍵），等待下一次玩家按方向鍵

### ▼行號 09

共執行 4 次迴圈區塊內的程式，迴圈內的預設變數 index 會從 0 逐次增加到 3

### ▼行號 10

複製「設定按鍵」自訂函式中的「計次」迴圈程式，重新顯示左側 4 個紅框箭頭

「判斷重疊」自訂函式：

呼叫此自訂函式需帶入二個角色參數，判斷二個角色是否在同一垂直位置（y），如果相同代表按鍵正確得分 +1，並將發射物角色刪除，最後再依據得分計算白色箭頭產生的速度。

### 程式碼說明

#### ▼行號 02

判斷所傳入二個角色的垂直位置（y）是否相同，如果相同則執行條件成立區塊內程式  
從事件積木指令拖曳出「mySprite」、「mySprite2」放入角色名稱參數中

#### ▼行號 03

得分值 +1

#### ▼行號 04

從重疊事件積木指令拖曳出「mySprite2」放入參數中，將發生重疊的白色箭頭髮射物刪除，避免持續執行重疊事件

#### ▼行號 05

播放「ba ding」音效

#### ▼行號 06

使用「 $1000 - (\text{得分} * 5)$ 」運算式，依「得分」計算「延時」變數內容，得分越高產生的白色箭頭越快

#### ▼行號 01

當程式中呼叫自訂函式時執行函式區塊內的程式，呼叫此函式必須帶入二參數，分別是玩家按下的方向角色和發生重疊的發射物角色，將此二角色分別存入「mySprite」變數和「mySprite2」變數中



The image shows a Scratch code block for a custom function named "判斷重疊" (Check Overlap). The function takes two arguments: "mySprite" and "mySprite2". The code block contains the following logic:

- 如果** (If) `mySprite y = mySprite2 y` **那麼** (Then):
  - 得分改變** (Score Change) `1`
  - 角色** (Sprite) `mySprite2` **摧毀** (Destroy) `+`
  - 播放音效** (Play Sound) `ba ding`
- 變數** (Variable) `延遲` (Delay) **設為** (Set to) `1000 - (得分 * 5)`

## 9.6 完整程式碼

〔程式檔：09 指尖跳舞〕

The image displays the Scratch code for the 'Finger Dance' game, organized into several functional blocks:

- 當啟動時 (When the game starts):**
  - 呼叫 設定圖形陣列 (Call 'Set Arrow Array')
  - 呼叫 繪製背景 (Call 'Draw Background')
  - 變數 方向 設為 4 (Set 'Direction' variable to 4)
  - 變數 延遲 設為 1000 (Set 'Delay' variable to 1000)
  - 生命設為 20 (Set 'Lives' to 20)
  - 對話內容 "按A開始遊戲" 版面位置 下 (Set dialog content to "Press A to start the game" at the bottom)
- 當 上 鍵 被 按住 (When the up key is pressed):**
  - 呼叫 設定按鍵 0 (Call 'Set Key 0')
- 當 左 鍵 被 按住 (When the left key is pressed):**
  - 呼叫 設定按鍵 1 (Call 'Set Key 1')
- 定義函式 設定圖形陣列 (Define function 'Set Arrow Array'):**
  - 陣列 (Array) 包含四個 '角色' (Sprite) 類別 'Player'。
  - 變數 箭頭角色 設為 (Set 'Arrow Sprite' variable to the array).
  - 變數 白色箭頭 設為 陣列 [白色, 左, 右, 下] 箭頭 (Set 'White Arrow' variable to the array of white arrows).
  - 變數 紅框箭頭 設為 陣列 [左, 右, 下, 上] 箭頭 (Set 'Red Box Arrow' variable to the array of red box arrows).
  - 變數 黃底箭頭 設為 陣列 [上, 左, 右, 下] 箭頭 (Set 'Yellow Arrow' variable to the array of yellow arrows).
- 定義函式 繪製背景 (Define function 'Draw Background'):**
  - 背景顏色設為 深藍色 (Set background color to dark blue).
  - 變數 畫布 設為 圖像創建 寬度 160 高度 55 (Set 'Canvas' variable to create image with width 160 and height 55).
  - 計次 index 從 0 到 3 (Loop 'index' from 0 to 3):
    - 角色 取得 箭頭角色 的項目值 索引值為 index 的位置設為 x 10 y index \* 16 + 32 (Get the sprite from 'Arrow Sprite' array at index 'index', position x=10, y=index\*16+32).
    - 角色 取得 箭頭角色 的項目值 索引值為 index 的圖像設為 取得 紅框箭頭 的項目值 索引值為 index (Get the image from 'Arrow Sprite' array at index 'index' and set it to the image from 'Red Box Arrow' array at index 'index').
    - 圖像 畫布 繪製線條 起點 x 0 y index \* 16 終點 x 160 y index \* 16 顏色 黃色 (Draw a line on the canvas from (0, index\*16) to (160, index\*16) in yellow).
  - 變數 線 設為 角色 畫布 類別 line (Set 'Line' variable to the 'Canvas' sprite category 'line').
  - set 線 z (深度) to -1 (Set the 'Line' variable's z-depth to -1).
- 當 右 鍵 被 按住 (When the right key is pressed):**
  - 呼叫 設定按鍵 2 (Call 'Set Key 2')
- 當 下 鍵 被 按住 (When the down key is pressed):**
  - 呼叫 設定按鍵 3 (Call 'Set Key 3')
- 重複無限次 (Repeat infinitely):**
  - 變數 列 設為 隨機取數 0 到 3 (Set 'Column' variable to a random number from 0 to 3).
  - 變數 隨機箭頭 設為 發射物 取得 白色箭頭 的項目值 索引值為 列 來自側邊 速度 vx -100 vy 0 (Set 'Random Arrow' variable to a projectile from the 'White Arrow' array at index 'Column', coming from the side with velocity vx=-100, vy=0).
  - set 隨機箭頭 y to 列 \* 16 + 33 (Set the 'Random Arrow' variable's y-position to Column \* 16 + 33).
  - 暫停 延遲 毫秒 (Pause for a delay in milliseconds).

程式碼續接下頁

```
定義函式 設定按鍵 num
  變數 方向 設為 num
  計次 index 從 0 到 3
  執行 角色 取得 箭頭角色 的項目值 索引值為 index 的圖像設為 取得 紅框箭頭 的項目值 索引值為 index
  角色 取得 箭頭角色 的項目值 索引值為 num 的圖像設為 取得 黃底箭頭 的項目值 索引值為 num
```

```
定義函式 判斷重疊 mySprite mySprite2
  如果 < mySprite y = mySprite2 y > 那麼
    得分改變 1
    角色 mySprite2 摧毀 +
    播放音效 ba ding
  變數 延遲 設為 1000 - 得分 x 5
```

```
當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Projectile 時
  如果 方向 < 4 > 那麼
    呼叫 判斷重疊 取得 箭頭角色 的項目值 索引值為 方向 otherSprite
  否則
    生命改變 -1
    播放音效 pew pew
    角色 otherSprite 摧毀 +
  變數 方向 設為 4
  計次 index 從 0 到 3
  執行 角色 取得 箭頭角色 的項目值 索引值為 index 的圖像設為 取得 紅框箭頭 的項目值 索引值為 index
```

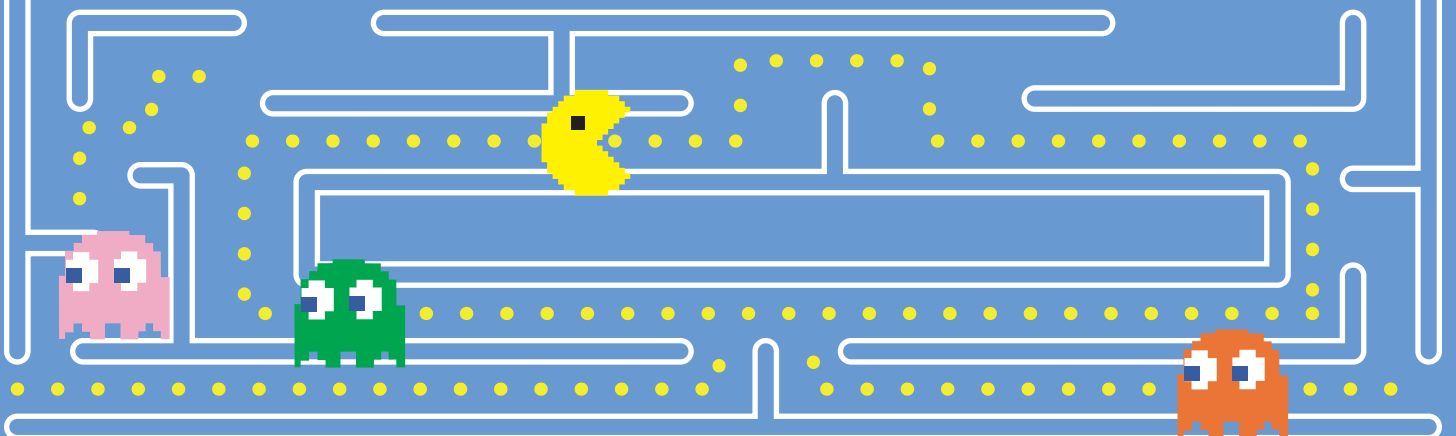
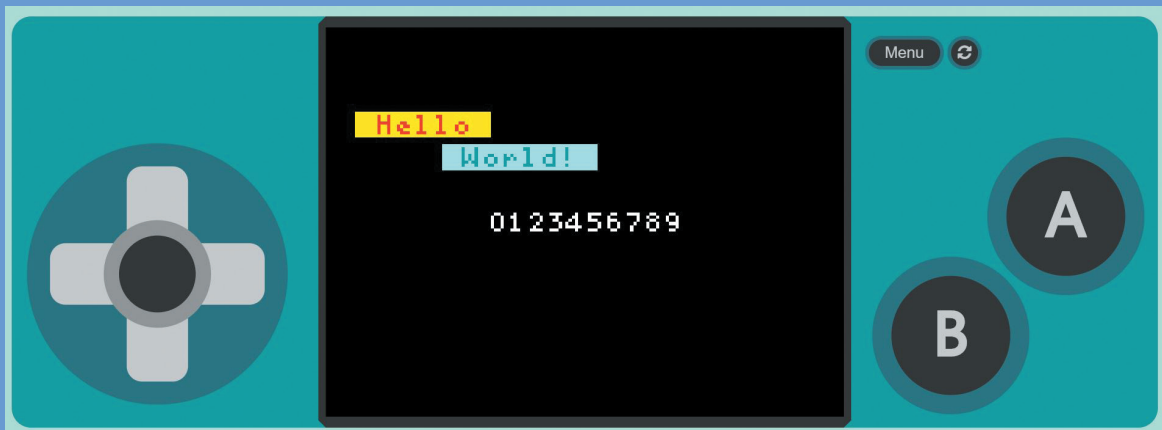
## 練習一下

目前的遊戲是白色箭頭出現後，不需等白色箭頭很靠近左側紅框箭頭，只要玩家所按的方向鍵正確時即可得分，請多加一個「當遊戲更新」事件，白色箭頭必須靠近玩家時，並且玩家所按的方向鍵正確時才能得分。

# 10 Chapter

## 自訂 Arcade 文字積木

MakeCode 允許使用者自訂積木，可將一小段完成的積木程式自訂成單一個積木指令或是自行撰寫一段 JavaScript 程式再自訂成一個積木指令，將常用的程式功能濃縮在一個積木指令中，方便未來取用。

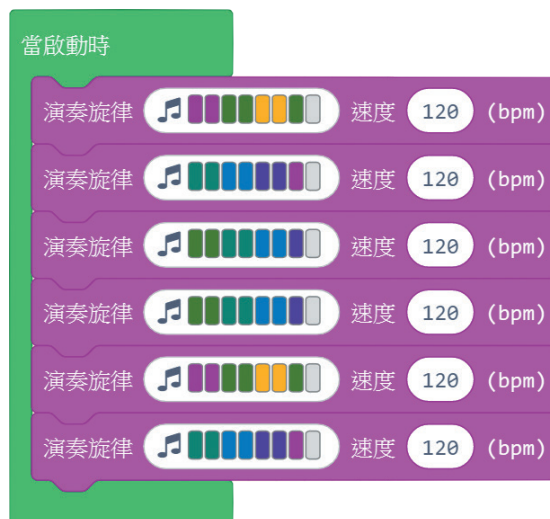


自訂積木較簡單的二種方式，第一種是將自訂積木程式新增一個檔案（檔名：xxx.ts）至「資源管理器」內，一旦完成自訂積木立刻可在積木盒中取用，這種方式最簡單但自訂積木需與專案一起儲存，無法獨立取出至其他專案使用；第二種是將自訂積木上傳至 GitHub 網站，如此，就能在任何時候使用擴展功能直接從網路下載自訂積木，方便提供給任一專案使用。

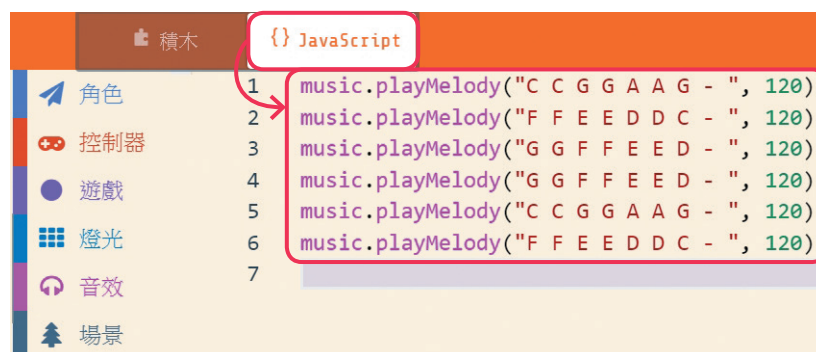
## 10.1 使用積木程式自訂積木指令

使用系統提供的積木，將一段設計好的演奏「小星星」積木程式自訂成單一個積木指令。

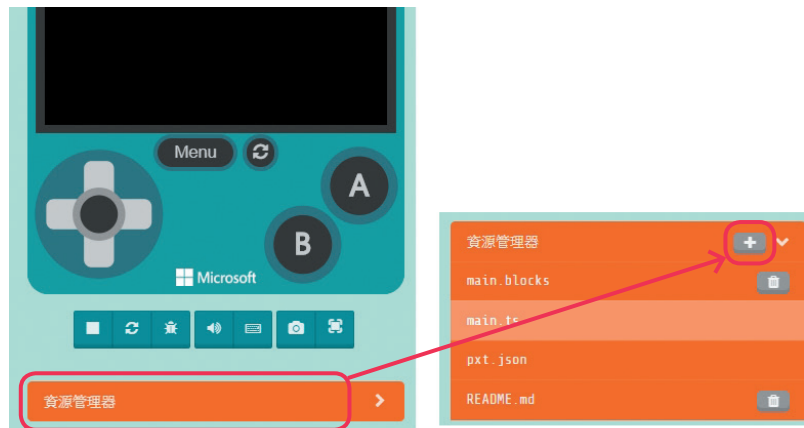
**step 1** 設計演奏「小星星」的程式放在「當啟動時」事件（參考附錄二、範例 16）



**step 2** 切換至「JavaScript」編輯器，將所有 JavaScript 程式碼剪下



**step 3** 點選左側「資源管理器」，再按「+」添加自訂積木



**step 4** 輸入積木自訂檔案名稱「LittleStar」，副檔名預設為「.ts」

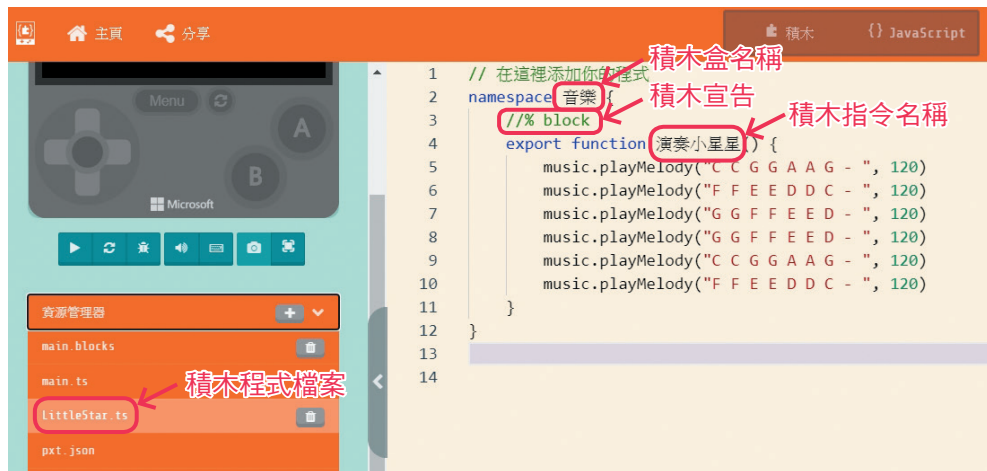


**step 5** 先在「JavaScript」編輯區輸入以下程式碼

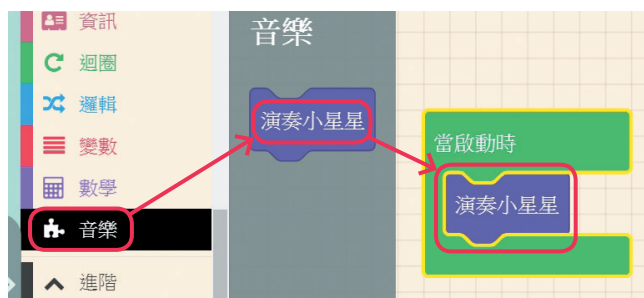
```
namespace 音樂 {
  //% block
  export function 演奏小星星 () {

  }
}
```

**step 6** 再將剪下之 JavaScript 程式碼貼在積木指令內層的大括號中，完成後如下：



**step 7** 切回積木編輯區，從「音樂」積木盒中拖曳出「演奏小星星」積木指令到「當啟動時」事件，即完成第一種自訂積木。



## 10.2 撰寫 JavaScript 自訂文字積木指令

Arcade 並無專用文字顯示積木，如果要顯示文字必須使用「讓角色說出」積木指令，但此積木指令固定為白底黑字，無法自行更改文字顏色和文字背景色，本章使用 JavaScript 寫一段文字顯示的功能，簡單製作成一個文字顯示積木。

**step 1** 切換至「JavaScript」編輯器

**step 2** 點選左側「資源管理器」，再按「+」添加自訂積木

**step 3** 輸入積木自訂檔案名稱「CreateTextSprite」，副檔名預設為「.ts」

**step 4** 在「JavaScript」編輯區輸入以下程式碼

程式碼說明

▼行號 03

積木區塊 (block)，後面可選擇加入積木上要顯示的標題文字及參數標題  
積木指令如有帶參數，則使用「參數標題 | % 參數」

▼行號 04

設定「fg」和「bg」參數可使用拉桿輸入的整數範圍

▼行號 05

設定「text」參數預設的文字內容為「Hello」

▼行號 06

使用「export function 積木指令名稱 ( ) { }」定義積木指令，一個積木盒中可有多個積木指令，積木指令如有帶參數，在小括號中宣告「參數：資料型態」，並使用逗號分隔參數

▼行號 07

設定文字字體大小，只有三種大小：5 號、8 號、12 號 (使用 image.fontXX)

▼行號 08

由於文字是使用圖片方式顯示於屏幕上，故設定圖片寬度為字元寬度 \* 字元長度

▼行號 09

設定圖片高度為字元高度

▼行號 10

建立圖片

▼行號 11

設定圖片填滿，填滿顏色為呼叫積木指令時傳入的第三個參數內容

▼行號 01

「//%」為積木設定的前綴字元，在前綴字元後面的屬性如下說明：  
「color」積木盒顏色，色碼可從 <https://www.ifreesite.com/color/> 取得  
「weight」積木盒排列位置，預設為 0 (積木區最下面)  
「icon」積木盒前的小圖案，圖案代碼可從 <https://fontawesome.com/> 取得 (代碼前加上 \u)

▼行號 02

使用「namespace 積木名稱 { }」指令，定義積木盒，所有積木指令程式都要放在大括號中

```
01 //% color="#FFAB19" weight=100 icon="\uf08c"
02 namespace 自訂文字 {
03     //% block="顯示 | 文字 %text | 顏色 %fg | 背景 %bg"
04     //% fg.min=0 fg.max=15 bg.min=0 bg.max=15
05     //% text.defl="Hello"
06     export function 顯示文字 (text: string, fg: number,
07     bg: number) {
08         const font = image.font8;
09         const width = font.charWidth * text.length;
10         const height = font.charHeight;
11         const res = image.create(width, height);
12         res.fill (bg) ;
13         res.print(text, 0, 0, fg, font);
14         const sprite = sprites.create(res, SpriteKind.
15         Food);
16         return sprite;
17     }
18 }
```

▼行號 12

將文字繪製到圖片上，參數 (文字，座標 X，座標 Y，文字顏色，字體大小)

▼行號 13

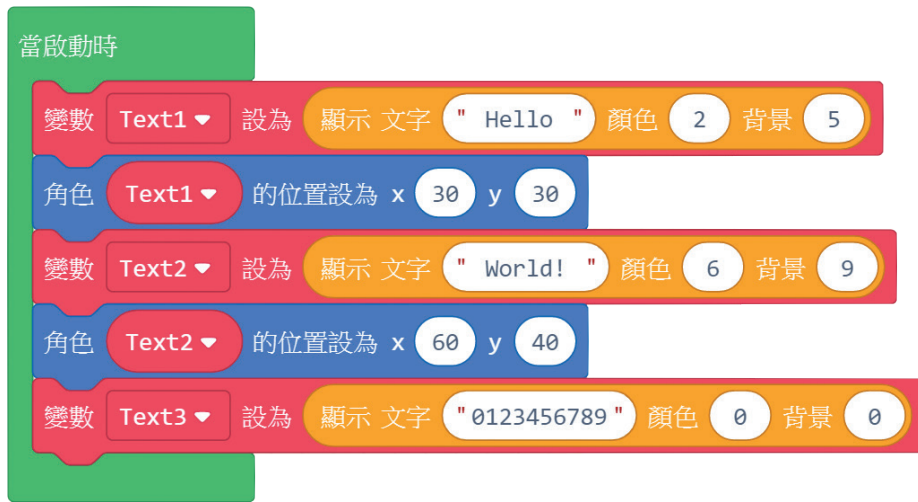
建立角色物件，並預設圖示為文字圖片；類別為 Food

▼行號 14

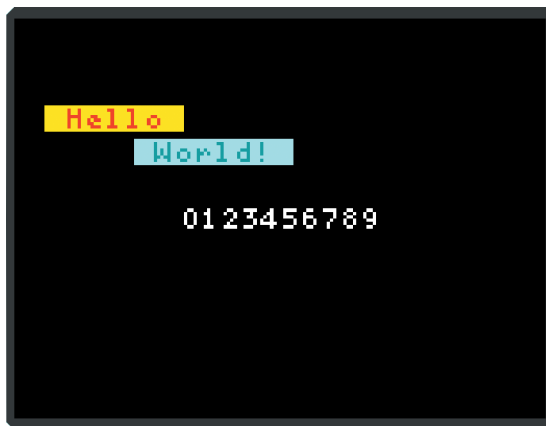
回傳角色物件

詳細製作積木語法可參考官方網頁：<https://makecode.com/defining-blocks>

**step 5** 切回積木編輯區，從「自訂文字」積木盒中拖曳出「顯示文字」積木指令到「變數設為 ..」積木指令中，程式如下：



**step 6** 執行結果

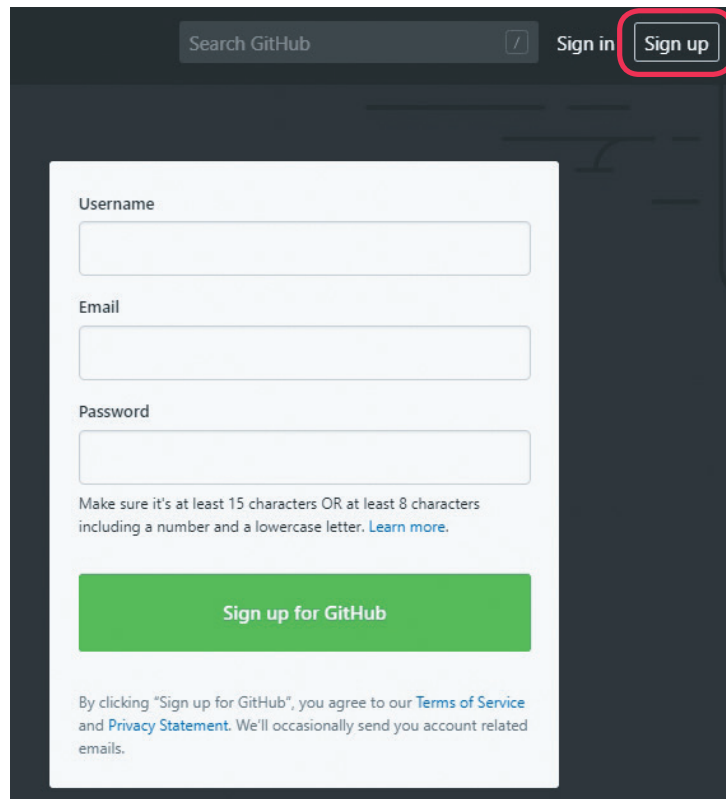


## 10.3 申請 GitHub 帳號

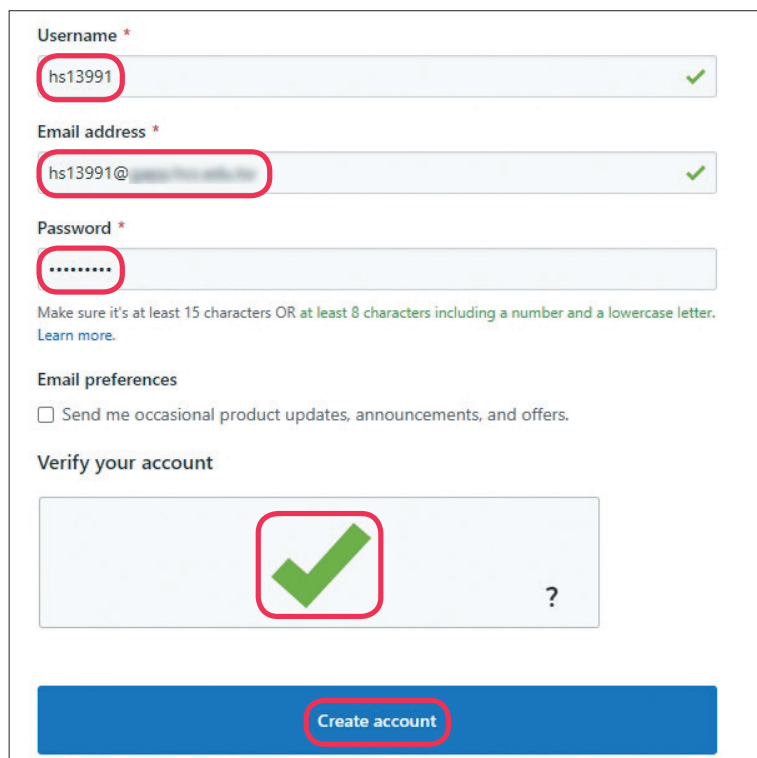
Git 是一個分散式版本控制軟體，Github 則是一個支援 Git 程式碼存取和遠端托管的平台服務，如果希望開發的積木能隨時下載使用，必須將積木文件上傳至自己申請的 GitHub 數據庫（Repository）中，帳號申請如下：

**step 1** 在瀏覽器網址輸入：<https://github.com/>

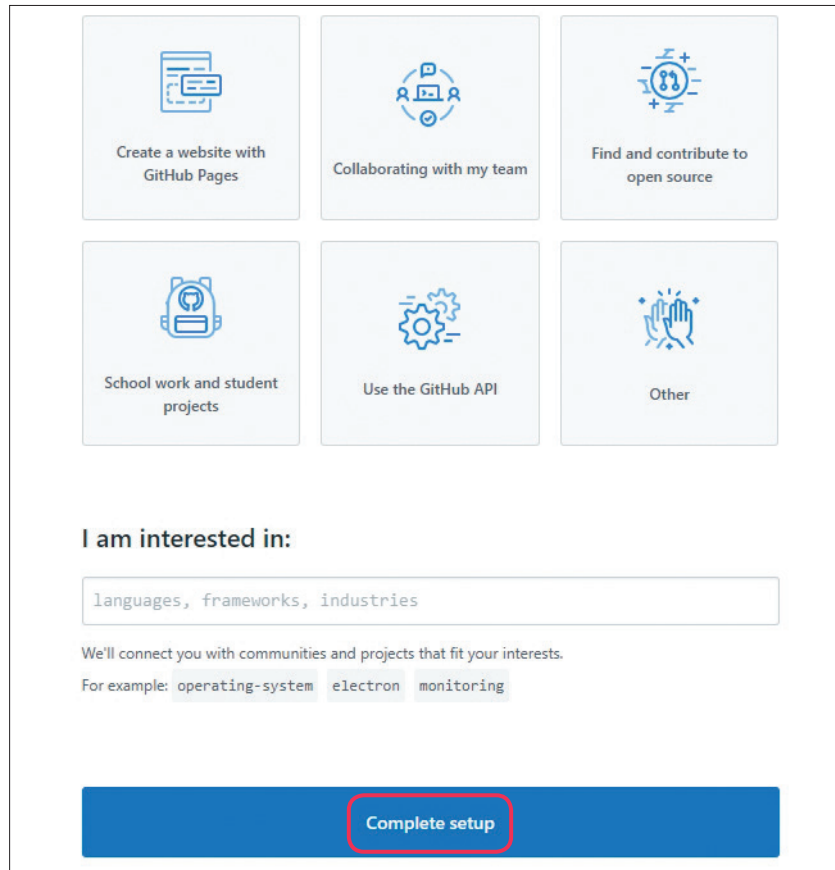
step 2 右上角點選「Sign up」（註冊）



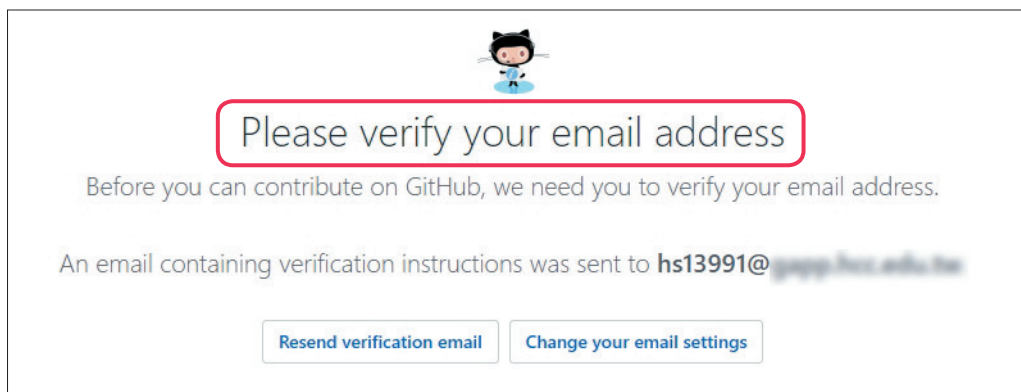
step 3 輸入自訂帳號、電子郵件、自訂密碼、通過驗證後（驗證時旋轉動物圖案，讓圖案轉正即可），點選「Create account」（建立帳號）



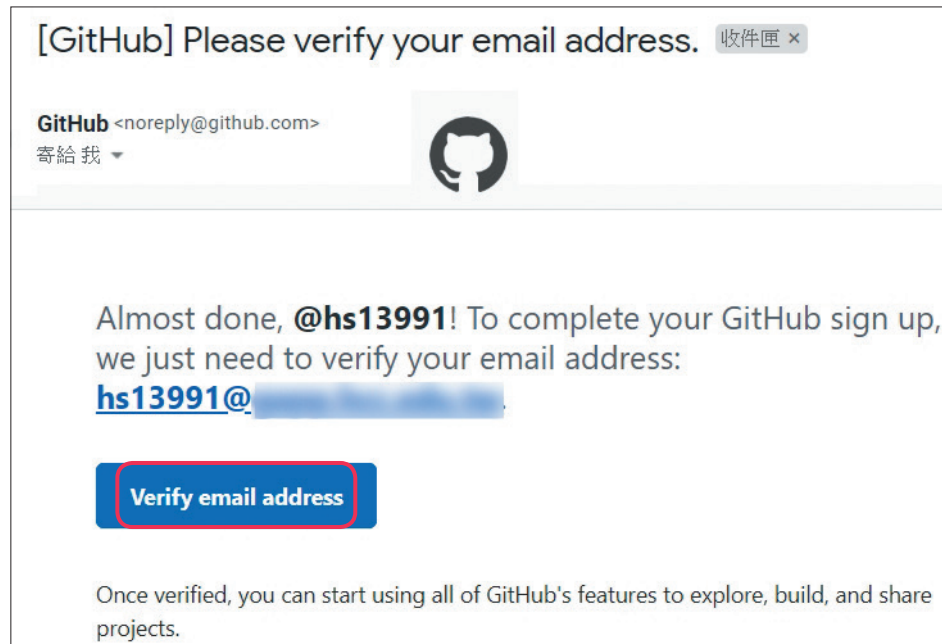
**step 4** 問卷網頁不需回答，直接拖曳網頁至最下方，點選「Complete setup」（完成設定）



**step 5** 已寄送驗證電子郵件通知網頁，請至電子郵件收驗證信（12）



**step 6** 開啟電子郵件驗證信，點選「Verify email address」（驗證郵件位址），即完成 GitHub 帳號註冊。



**step 7** 完成驗證後就會自動登入 GitHub 的網頁

## 10.4 製作自訂積木文件

MakeCode 自訂積木文件主要由三個文件檔案組成（main.ts、pxt.json、README.md），如果需要自動翻譯成中文時再多一個文件檔案（xxxx-strings.json），其中 xxxx 為自訂積木盒名稱，本範例使用「createText」，各檔案說明如下：

### 一、main.ts 文件檔案內容（定義積木外觀及內容的主要檔案）

#### 程式碼說明

##### ▼行號 03

積木區塊（block），後面可選擇加入積木上要顯示的標題文字及參數標題  
積木指令如有帶參數，則使用「參數標題 | % 參數」

##### ▼行號 04

設定「fg」和「bg」參數可使用拉桿輸入的整數範圍

##### ▼行號 05

設定「text」參數預設的文字內容為「Hello」

##### ▼行號 06

使用「export function 積木指令名稱（）{ }」定義積木函式，一個積木盒中可有多個積木函式，積木函式如有帶參數，在小括號中宣告「參數：資料型態」，並使用逗號分隔參數

##### ▼行號 07

設定文字字體大小，只有三種大小：5 號、8 號、12 號（使用 image.fontXX）

##### ▼行號 08

由於文字是使用圖片方式顯示於屏幕上，故設定圖片寬度為字元寬度 \* 字元長度

##### ▼行號 09

設定圖片高度為字元高度

##### ▼行號 10

建立圖片

##### ▼行號 11

設定圖片填滿，填滿顏色為呼叫積木指令時傳入的第三個參數內容

##### ▼行號 02

使用「namespace 積木盒名稱 { }」區塊，定義積木盒，所有積木指令程式都要放在大括號中

##### ▼行號 01

block="createText" 此為英文積木盒名稱，當使用繁體中文時，會以中文語系檔案（xxxx-strings.json）內的中文取代

```

01 //% color="#FFAB19" weight=100 icon="\uf08c" block="createText"
02 namespace createText {
03     //% block="disp|text %text |fgColor %fg | bgColor %bg"
04     //% fg.min=0 fg.max=15 bg.min=0 bg.max=15
05     //% text.defl="Hello"
06     export function dispText(text: string, fg: number, bg:
07         number) {
08         const font = image.font8;
09         const width = font.charWidth * text.length;
10         const height = font.charHeight;
11         const res = image.create(width, height);
12         res.fill(bg);
13         res.print(text, 0, 0, fg, font);
14         const sprite = sprites.create(res, SpriteKind.Food);
15         return sprite;
16     }

```

##### ▼行號 12

將文字繪製到圖片上，參數（文字，座標 X，座標 Y，文字顏色，字體大小）

##### ▼行號 13

建立角色物件，並預設圖示為文字圖片；類別為 Food

##### ▼行號 14

回傳角色物件

## 二、pxt.json 文件檔案內容（定義與積木相關的所有檔案）

### 程式碼說明

▼行號 02

「"name"」後面的「"createText"」  
為自訂積木盒名稱

▼行號 03

版本宣告

▼行號 04

描述積木功能

▼行號 09

「"files"」下面宣告積木會用到的  
文件檔案

▼行號 10

積木主要程式

▼行號 11

在 GitHub 網頁中顯示的積木說明  
文件

▼行號 12

「createText-strings.json」為多國語  
系檔，需存放於「\_locales/zh」資  
料夾中，其中 createText 為自訂積  
木盒名稱

```

01 {
02   "name": "createText",
03   "version": "0.0.1",
04   "description": "自訂文字積木",
05   "license": "MIT",
06   "dependencies": {
07     "core": "*"
08   },
09   "files": [
10     "main.ts",
11     "README.md",
12     "_locales/zh/createText-strings.json"
13   ],
14   "public": true
15 }

```

### 三、README.md 文件檔案內容（在 GitHub 顯示的積木說明文件）

#### 程式碼說明

##### ▼行號 01

「#」後面的「"createText"」為自訂積木盒名稱

##### ▼行號 02

在 GitHub 網頁中顯示的積木說明

##### ▼行號 12

「"createText"」為自訂積木盒名稱  
 「"hs13991"」為 GitHub 帳號  
 「"pxt-arcade"」為放置的數據庫名稱，需以「pxt-」為開頭

```

01 #createText
02 自訂 Arcade 文字積木
03
04 ## License
05 MIT
06
07 ## Supported targets
08 * for PXT/arcade
09 (The metadata above is needed for package search.)
10
11 ```package
12 createText=github:hs13991/pxt-arcade
13 ```
  
```

### 四、createText-strings.json 文件檔案內容

（多國語系檔案，中文語系檔案存放在「\_locales」資料夾中的「zh」資料夾內）

#### 程式碼說明

##### ▼行號 02

「"createText"」為自訂積木盒英文名稱  
 「"自訂文字"」為自訂積木盒中文名稱

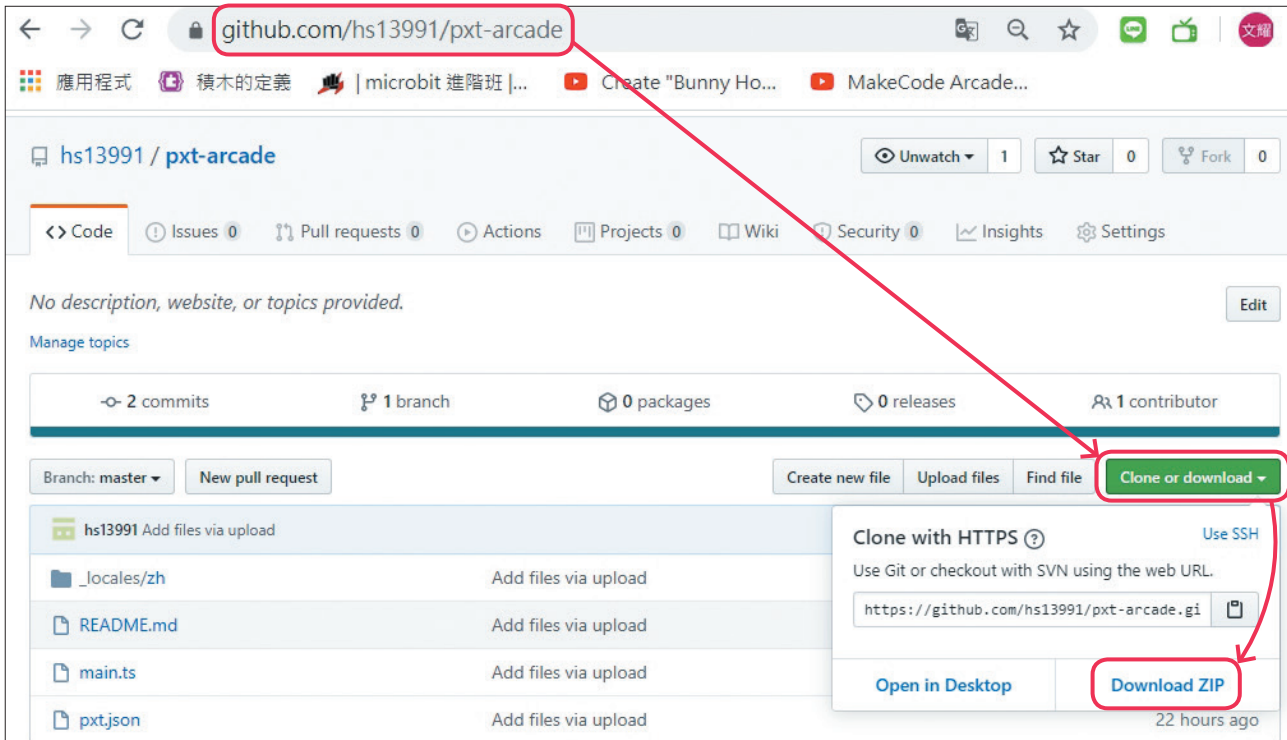
```

01 {
02   "createText|block": "自訂文字",
03   "createText.dispText|block": "顯示 文字 %text| 顏色 %fg| 背景 %bg"
04 }
  
```

##### ▼行號 03

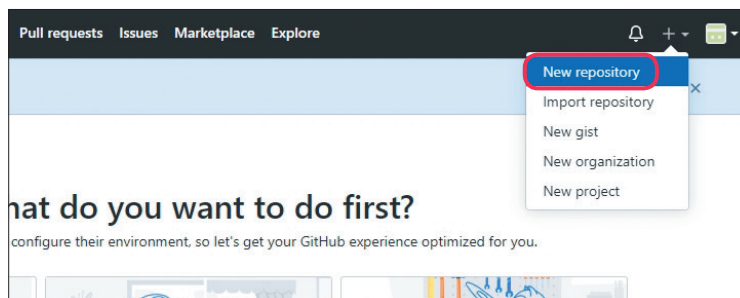
「"createText.dispText"」為自訂積木指令英文名稱  
 「"顯示 文字 %text| 顏色 %fg| 背景 %bg"」為自訂積木指令對應的中文參數標題

上述四個文件檔案可從 <https://github.com/hs13991/pxt-arcade>，點選「Clone or download」再選擇「Download ZIP」下載到電腦。

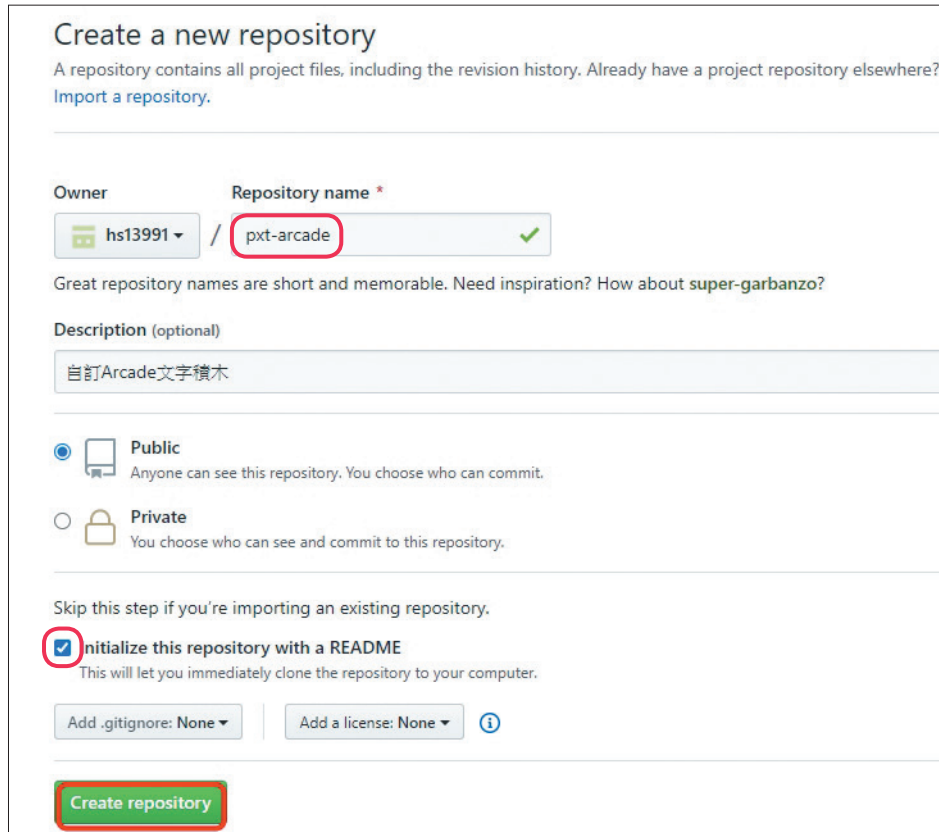


## 10.5 上傳自訂積木文件

**step 1** 登入 GitHub 網頁後，點選右上角「+」，下拉選單選擇「New repository」（建立數據庫）



**step 2** 輸入數據庫名稱「pxt-arcade」（數據庫名稱需以「pxt-」為開頭）；勾選「Initialize this repository with a README」使用上傳的說明檔初始化數據庫；再按「Create repository」建立數據庫



**Create a new repository**  
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: hs13991 / Repository name: **pxt-arcade** ✓

Great repository names are short and memorable. Need inspiration? How about [super-garbanzo?](#)

Description (optional): 自訂Arcade文字積木

**Public**  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

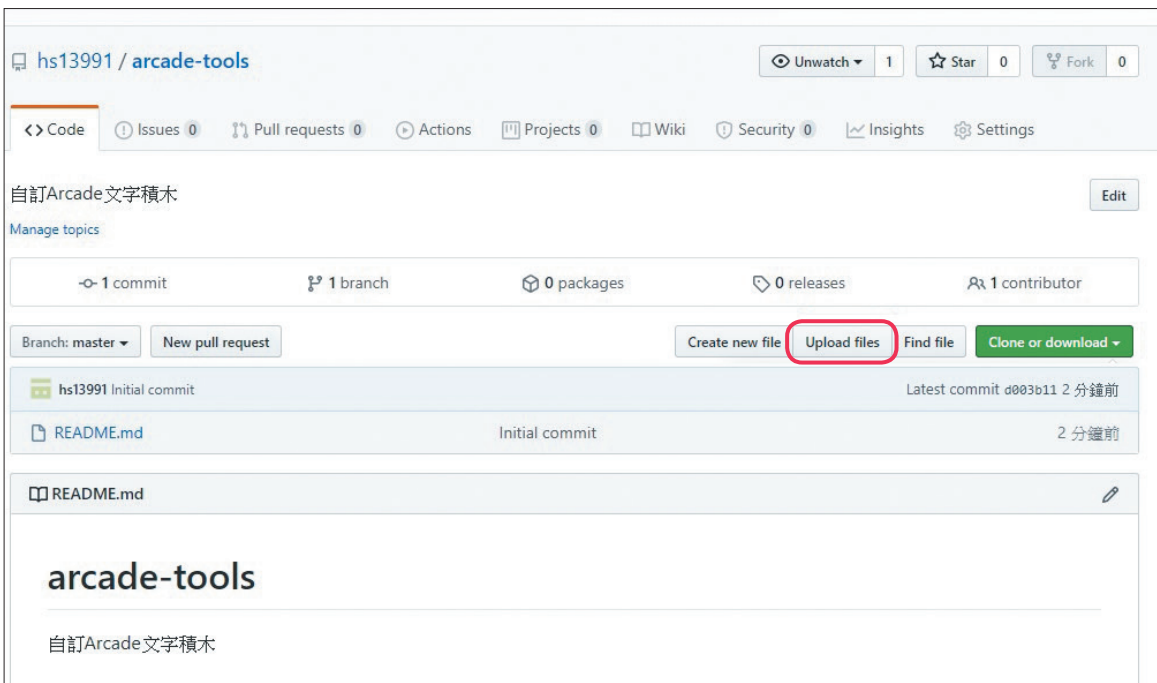
Skip this step if you're importing an existing repository.

**Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

**Create repository**

**step 3** 按「Upload Files」開啟檔案上傳網頁



hs13991 / arcade-tools

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights | Settings

自訂Arcade文字積木 Edit

Manage topics

1 commit | 1 branch | 0 packages | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | **Upload files** | Find file | Clone or download

hs13991 Initial commit | Latest commit d003b11 2分鐘前

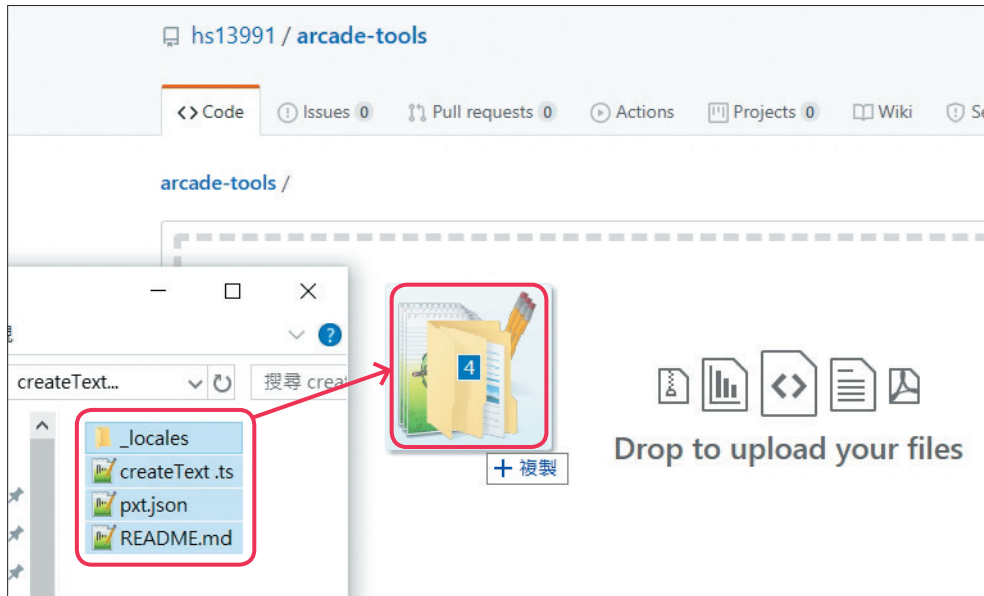
README.md | Initial commit | 2分鐘前

README.md Edit

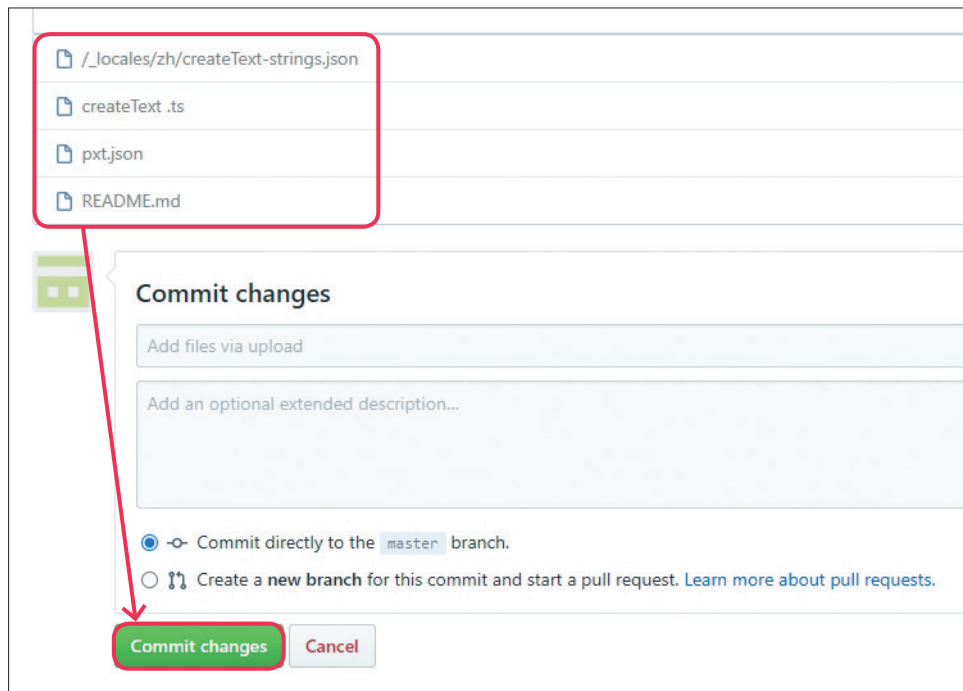
# arcade-tools

自訂Arcade文字積木

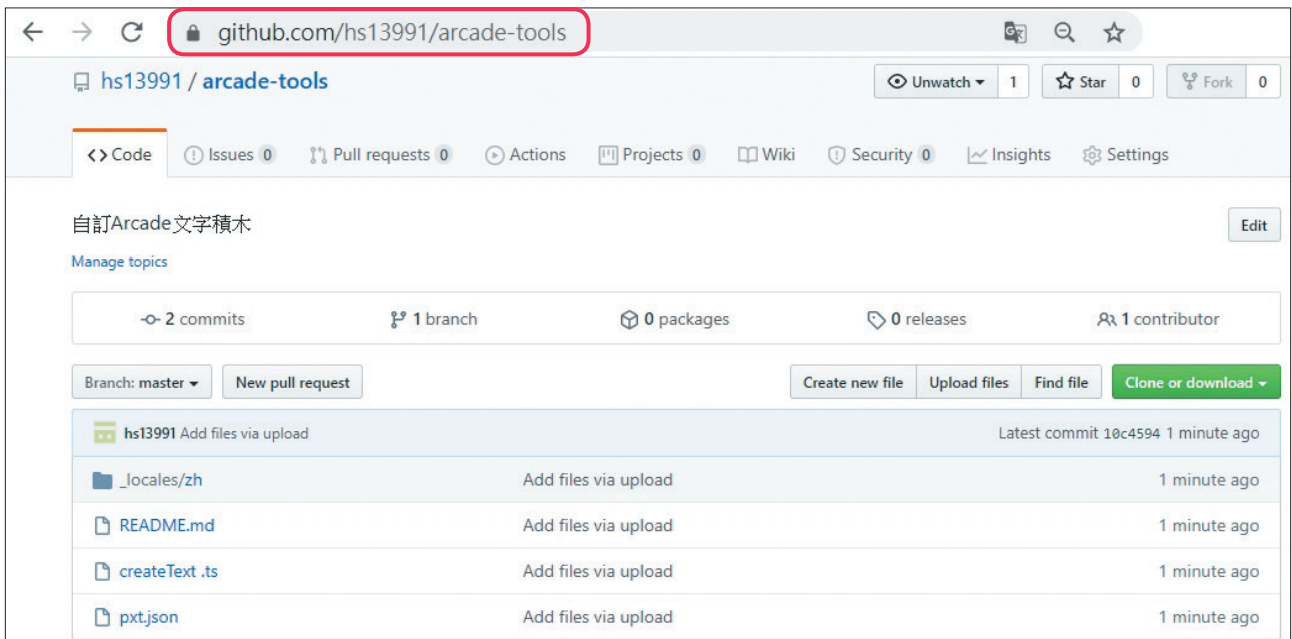
step 4 將建立好的三個積木文件和資料夾拖曳至檔案上傳區



step 5 上傳完畢後，按「Commit changes」提交變更



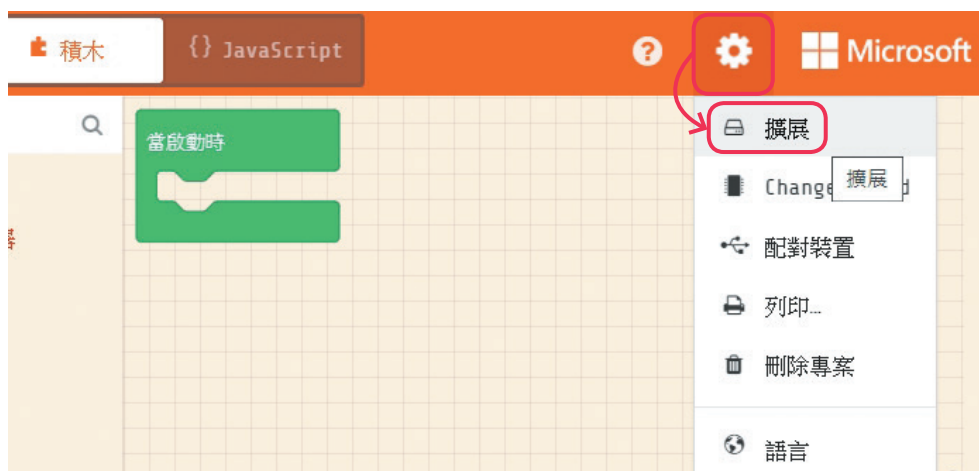
**step 6** 檔案提交完成，將網址複製起來，作為擴展自訂積木時的下載網址



## 10.6 使用擴展自訂積木

將自訂積木文件檔案上傳至 GitHub 後，即可透過網址，隨時下載自訂積木，不需再與專案綁在一起，下載積木使用擴展功能：

**step 1** 點選程式編輯區右上角「齒輪」圖示，下拉選單選擇「擴展」



**step 2** 在搜尋欄貼上 GitHub 自訂積木網址「<https://github.com/hs13991/pxt-arcade>」，按「Enter」鍵；再點選「createText」自訂積木



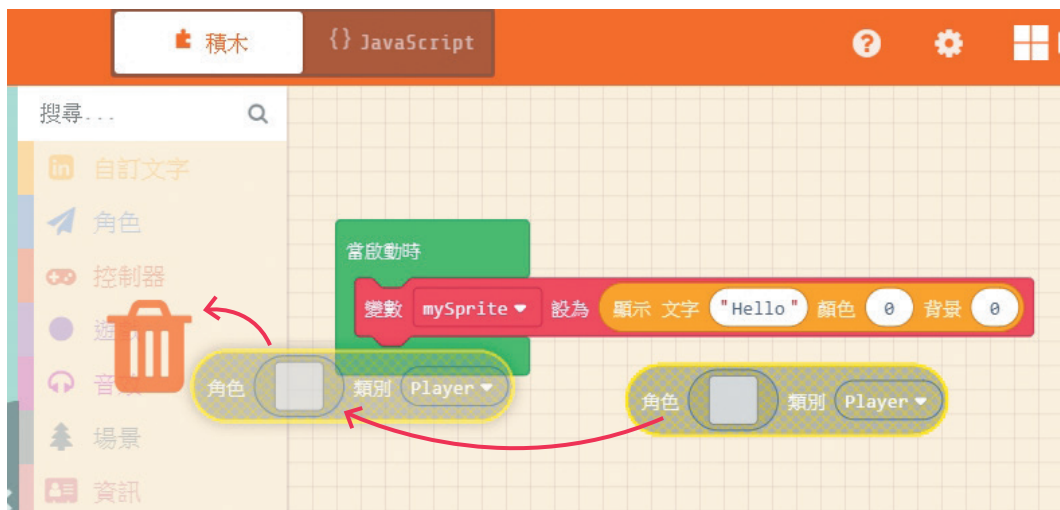
**step 3** 使用「角色」積木盒中的「變數設為角色」積木指令，拖曳到「當啟動時」事件積木中



**step 4** 使用「自訂文字」積木盒中的「顯示文字」積木指令，拖曳到「變數設為角色」積木指令中的角色積木，取代角色積木



**step 5** 將多餘的角色積木拖曳至積木區刪除



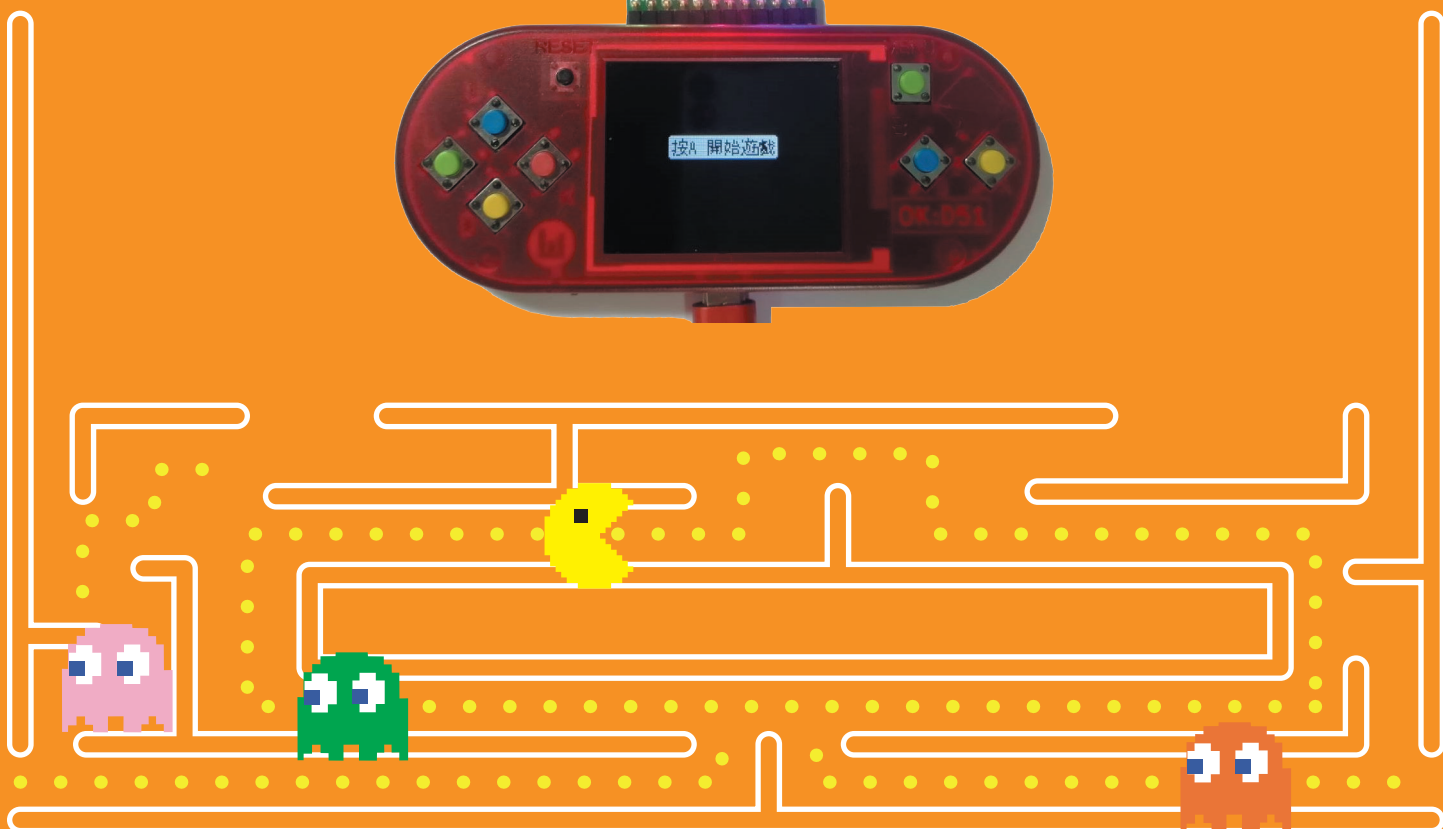
**step 6** 完成自訂積木結果：



# 11 Chapter

## 「電子輪盤」遊戲

WiFiBoy Arcade 遊戲機提供 12 腳位的擴充母座，可做為連接外部感測器或控制外部設備，本章將使用環形彩色 LED 燈板練習遊戲機外部控制，燈板資訊請參考 WiFiBoy Arcade 官方網站 <https://wifiboy.org/arcade/>



## 遊戲說明

利用環形彩色 LED 燈板上 12 顆全彩 LED 製作電子輪盤，按「A」鍵時其中一個燈開始快速順時針跑動，然後速度逐漸變慢最後停住，屏幕也會顯示相對應的 LED 編號。

## 01. 遊戲設計分析

### 一、遊戲角色：

角色名稱	角色類別	素材	創建角色的事件
顯示數字	Player	無	當啟動時

### 二、遊戲事件：

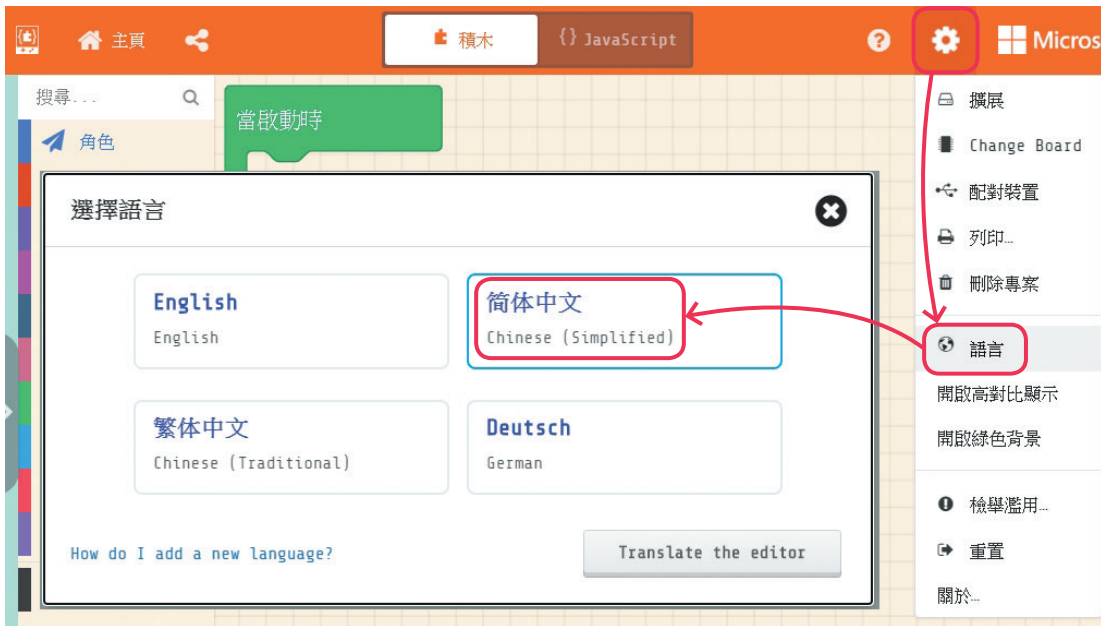


### 三、WiFiBoy Arcade 腳位

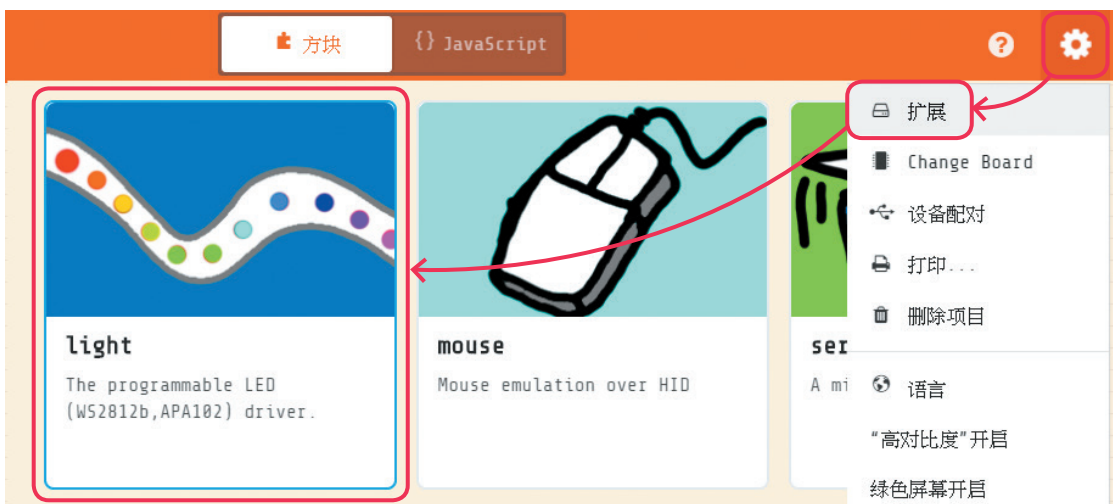
編號	主機腳位	燈板腳位
1	VUSB	5V
2	GND	GND
3	+3.3V	3V3
4	TX (A17)	DIN
5	RX (A16)	
6	MOSI (A0)	
7	MISO (B23)	
8	SCK (A1)	
9	SCL (A13)	
10	SDA (A12)	
11	SWDIO	
12	SWCLK	

## 11.2 「當啟動時」事件

由於繁體中文的「light」（燈光）積木盒開啟有錯誤（無法開啟「更多」積木選項），請先將語言切至簡體或英文，程式完成後再切回繁體中文即可。



使用「擴展」將「light」（燈光）積木盒載入至專案中。



## 程式碼說明

### ▼行號 03

創建燈板物件及設定相關參數

- ① 點選「燈泡」(燈光)積木盒後，再選擇「...更多」
- ② 拖曳出「將... 設為初始化燈帶」，變數名稱改為「燈板」
- ③ 下拉引腳選項選擇「TX」(傳送腳位)
- ④ 長度改為燈板提供的「12」顆全彩 LED



### ▼行號 01

當啟動時執行事件區塊內的程式

### ▼行號 02

在屏幕上創建一個角色，用於顯示目前亮燈編號

- ① 使用「精靈」(角色)積木盒中的「將變量設為精靈」積木指令
- ② 變數名稱改為「顯示數字」，不需設定圖案

### ▼行號 04

讓環型燈板顯示彩虹顏色

- ① 點選「燈泡」積木盒後，再選擇「...更多」
- ② 使用「動畫特效」(顯示動畫)積木指令
- ③ 變數名稱下拉選擇「燈板」並下拉「延時」(持續)選項選擇「2」秒

### ▼行號 05

設定環型燈板亮度 (0~255)

- ① 點選「燈泡」積木盒後，再選擇「...更多」
- ② 使用「設置.. 亮度」積木指令，亮度輸入「20」

### ▼行號 06

電子輪盤預設從 LED 0 開始點亮

- ① 使用「變量」(變數)積木盒中的「設置變量」
- ② 新增一個「燈編號」變數，將變數內容設定為 0，此變數內容為目前點亮的 LED 編號，LED 編號為 0~11

### ▼行號 07

屏幕顯示「按 A 開始遊戲」

- ① 使用「精靈」(角色)積木盒中的「精靈說...」積木指令
- ② 輸入「按 A 開始遊戲」



## 11.3 「當 A 鍵被按住」事件

### 程式碼說明

#### ▼行號 03

依「燈編號」變數內容指定點亮環形燈板中的 LED，顏色為綠色

- ① 點選「燈泡」積木盒後，再選擇「...更多」
- ② 使用「設置像素..顏色為」積木指令，變數名稱選擇「燈板」
- ③ 從「變量」積木盒中拖曳出「燈編號」變數至像素參數中
- ④ 點選顏色選項，選擇「綠色」

#### ▼行號 04

設定電子輪盤燈號移動的初始速度

- ① 使用「變量」積木盒中的「將變量設為」積木指令
- ② 下拉變數名稱選擇「延時」，變數內容設為「20」

#### ▼行號 05

讓 LED 先快速順時針移動 50 次

- ① 使用「循環」積木盒中的「重複 4 次」積木指令
- ② 將「4」改為「50」

#### ▼行號 06

呼叫點亮 LED 函式，函式內將依「燈編號」亮燈並順時針移動一次

- ① 使用「函數」（函式）積木盒中的「創建一個函數」
- ② 建立名為「亮燈」函式
- ③ 從「函數」積木盒拖曳出剛建立的函式呼叫指令

#### ▼行號 07

讓 LED 隨機再順時針移動 20~50 次，每次 index 變數內容增加 1

- ① 使用「循環」積木盒中的「對於從 0 至 4 的 index」積木指令
- ② 將結束值 4 改為使用「數學」積木盒中的「選取隨機數」積木指令
- ③ 隨機數範圍從「20」到「50」，

#### ▼行號 08

使用「函數」積木盒中的「呼叫亮燈」函式指令

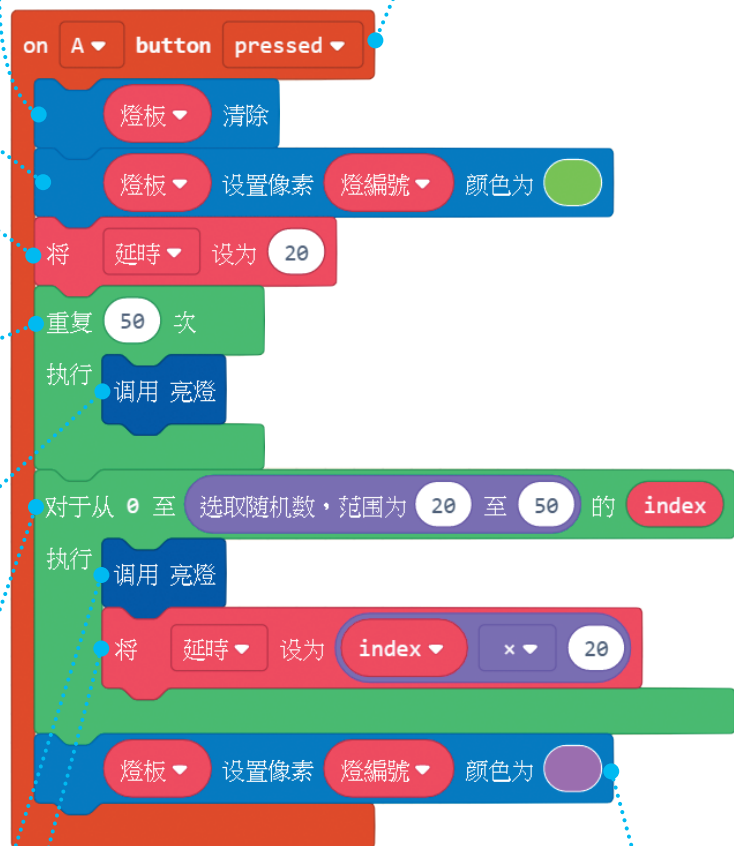
#### ▼行號 02

每次執行前先將目前的亮燈清除

- ① 點選「燈泡」積木盒後，再選擇「...更多」
- ② 使用「清除」積木指令，變數名稱選擇「燈板」

#### ▼行號 01

當按「A」時執行事件區塊內的程式



#### ▼行號 09

設定電子輪盤 LED 燈逐漸慢下來

- ① 使用「變量」積木盒中的「將變量設為」積木指令
- ② 下拉變數名稱選擇「延時」
- ③ 變數內容使用「數學」積木盒中的「乘法」積木指令
- ④ 使用運算式「index \* 20」，讓 index 內容值越大則延時越久

#### ▼行號 10

當電子輪盤停止後，讓最後的燈號從綠色改變成粉紅色

- ① 點選「燈泡」積木盒後，再選擇「...更多」
- ② 使用「設置像素..顏色為」積木指令，變數名稱選擇「燈板」
- ③ 從「變量」積木盒中拖曳出「燈編號」變數至「設置像素」參數中
- ④ 點選顏色選項，選擇「粉紅色」

「亮燈」自訂函式：

### 程式碼說明

#### ▼行號 03

因為只有 12 顆 LED 燈，「燈編號」變數內容固定為 0 ~ 11

- ① 使用「變量」積木盒中的「將變量設為」積木指令
- ② 下拉變數名稱選擇「燈編號」
- ③ 變數內容使用「數學」積木盒中的「相除的餘數」積木指令
- ④ 運算式為「燈編號 ÷ 12 的餘數」

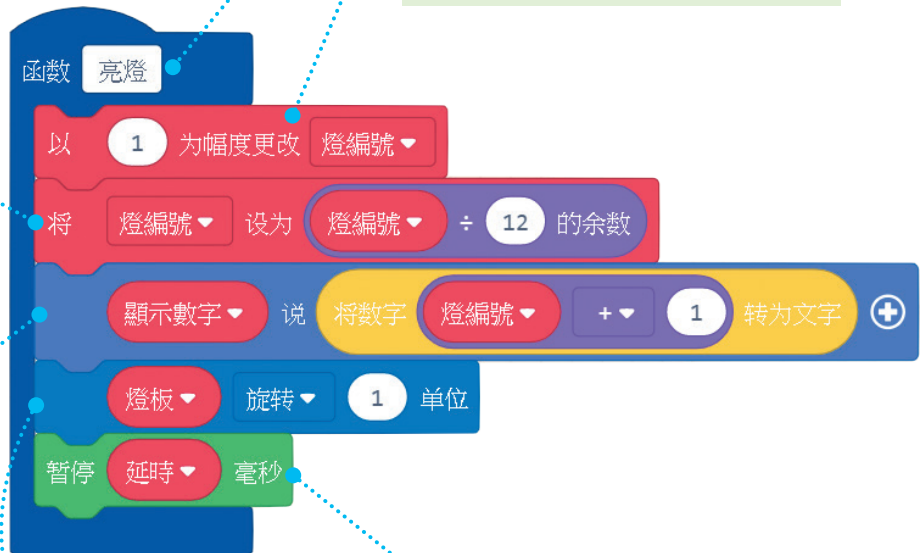
#### ▼行號 01

當程式中呼叫自訂函式時執行函式區塊內的程式

#### ▼行號 02

讓「燈編號」變數內容 +1

- ① 使用「變量」積木盒中的「以 1 為幅度更改」積木指令
- ② 下拉變數名稱選擇「燈編號」



#### ▼行號 04

讓屏幕中的「顯示數字」角色，說出目前點亮的 LED 燈編號

- ① 使用「精靈」（角色）積木盒中的「精靈說...」積木指令
- ② 說出內容使用「文本」（文字）積木盒中的「將數字轉為文字」積木指令
- ③ 說出內容為「燈編號」+1

#### ▼行號 05

讓 LED 燈順時針旋轉移動一次

- ① 點選「燈泡」積木盒後，再選擇「...更多」
- ② 使用「旋轉 1 位」積木指令

#### ▼行號 06

讓電子輪盤目前的亮燈暫停一下

- ① 使用「循環」積木盒中的「暫停」積木指令
- ② 從「變量」積木盒中拖曳出「延時」變數放到參數中

## 11.4 「當 B 鍵被按住」事件

### 程式碼說明

使用「遊戲」積木盒中的「遊戲復位」積木指令，當按「B」鍵時重新執行程式。



## 11.5 完整繁體中文程式碼

( 程式檔：11 電子輪盤 )

```

當啟動時
  變數 顯示數字 設為 角色 類別 Player
  變數 燈板 設為 燈條 WS2812 創建 引腳 TX 像素點 12
  燈條 燈板 顯示動畫 持續 2000 毫秒
  燈條 燈板 亮度設為 20
  變數 燈編號 設為 0
  角色 顯示數字 說出 "按A開始遊戲"

定義函式 亮燈
  變數 燈編號 改變 1
  變數 燈編號 設為 燈編號 ÷ 12 的餘數
  角色 顯示數字 說出 轉換 燈編號 + 1 成文字型別
  燈條 燈板 移動 旋轉 1 點
  暫停 延時 毫秒

on A button pressed
  燈板 清除
  燈板 設置像素 燈編號 顏色為 綠色
  將 延時 設為 20
  重複 50 次
    執行 調用 亮燈
  對於從 0 至 選取隨機數，範圍為 20 至 50 的 index
    執行 調用 亮燈
    將 延時 設為 index × 20
  燈板 設置像素 燈編號 顏色為 紫色

當 B 鍵被 按住
  遊戲重設
  
```

### 練習一下

請使用環形燈板設計燈號（左半 LED 綠；右半 LED 紅）交換閃爍的警示燈效果。

# 綜合實作題

**題目名稱** 小鴨鴨遊戲

**題目說明** 請以「第 5 章離線小鴨鴨」程式為基礎，在障礙物出現前 0.1 秒，讓環形彩色 LED 全亮紅色，做為提示燈號（障礙物出現後環形彩色 LED 熄滅）

## 解題提示

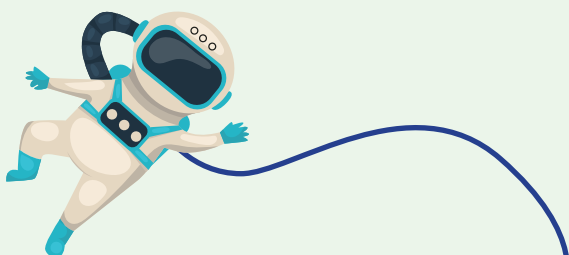
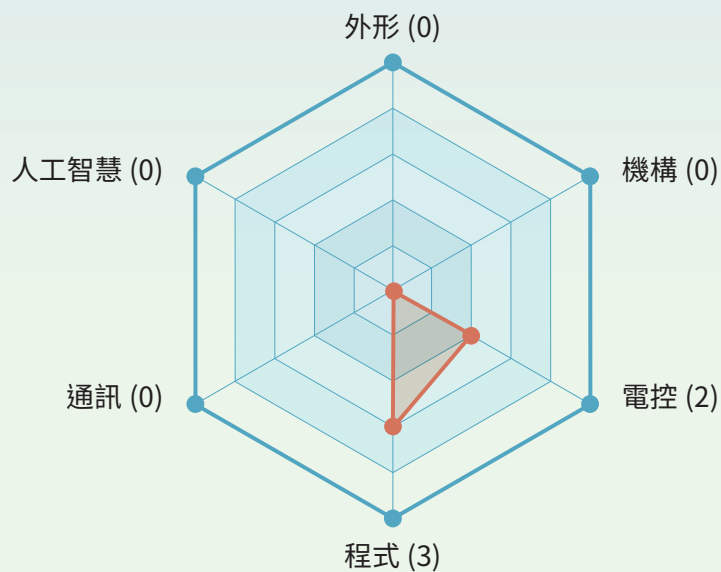
由於在障礙物出現前，須讓環形彩色 LED 全亮紅色 0.1 秒，但在「當遊戲更新」事件中，如果使用「暫停」指令，遊戲將會暫停執行，故應新增一個旗號變數，當障礙物要出現時，將旗號變數設定為 1，並在「重複無限次」事件中持續判斷旗號變數如果為 1，則讓環形彩色 LED 全亮紅色，0.1 秒後環形彩色 LED 熄滅，並產生障礙物。

創客題目編號：A033003

## 創客指標

外形	0
機構	0
電控	2
程式	3
通訊	0
人工智慧	0
<b>創客總數</b>	<b>5</b>

45  
mins



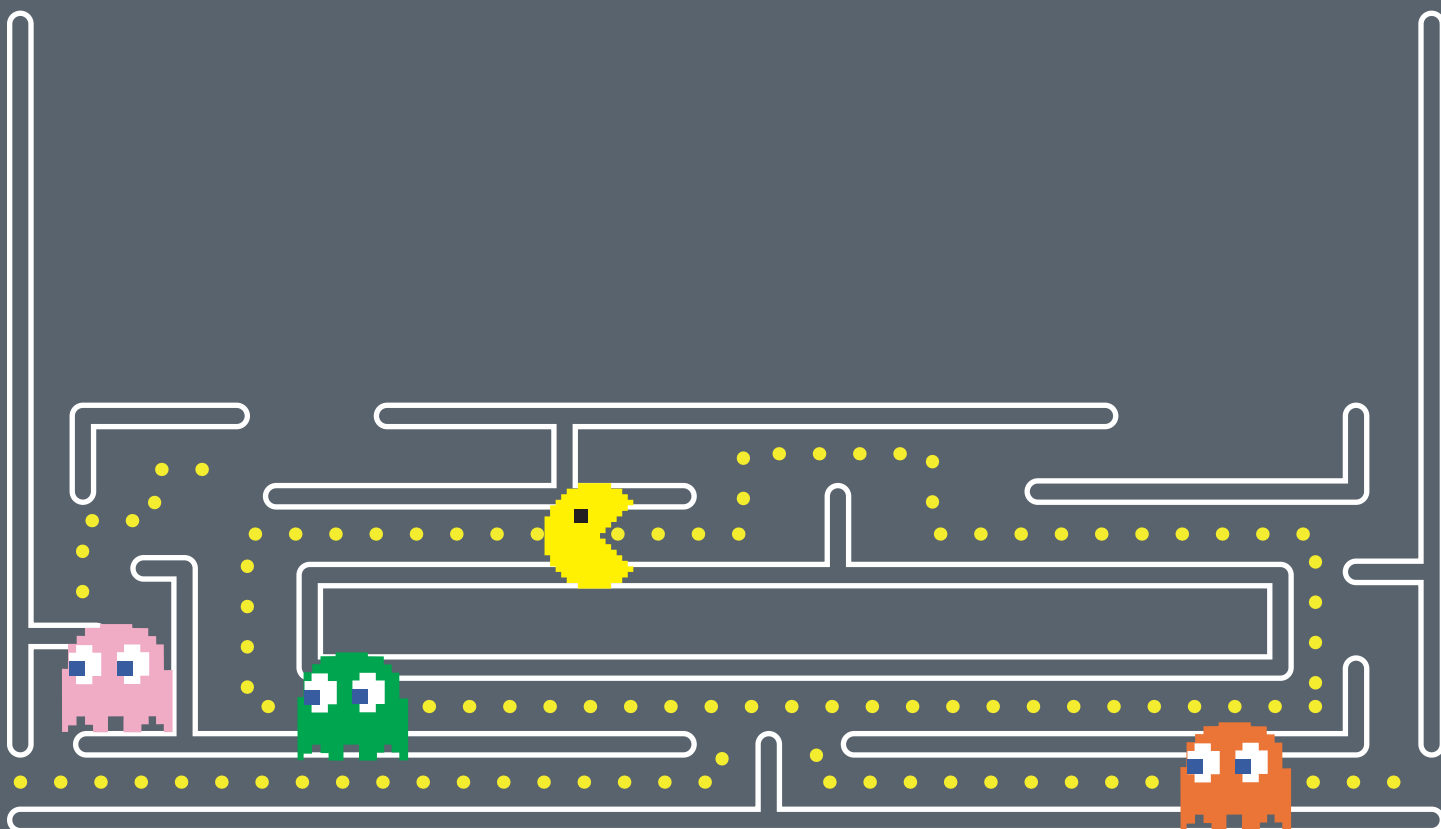


# 附錄

附錄一、各章練習參考解答

附錄二、Arcade 常用積木指令範例

附錄三、Arcade 積木指令介紹



## 附錄一、各章練習參考解答

### 練習一下

#### 第二章



```
當遊戲更新
  如果 隨機取布林值 true 的機率設為 25 % 那麼
    變數 雪花 設為 發射物 來自側邊 速度 vx -30 vy 30
    角色 雪花 的位置設為 x 隨機取數 0 到 畫面寬度 y 0
    角色 雪花 標記 變成幻影 設為 開
```

#### 第三章



```
當 B 鍵被 按住
  變數 projectile 設為 發射物 來自 ship 速度 vx 0 vy -140
  角色 projectile 開始效果 冷色放射狀 (cool radial) 持續 100 毫秒
```

解析度設為16x8

#### 第四章



```
當遊戲更新
  如果 得分 > 30 那麼
    變數 延時 設為 400
  否則如果 得分 > 20 那麼
    變數 延時 設為 600
  否則如果 得分 > 10 那麼
    變數 延時 設為 800
  否則
    變數 延時 設為 1000
```

## 第五章



## 第六章



## 第七章

```

當遊戲更新 間隔 500 毫秒
    變數 飛碟 設為 發射物 來自側邊 速度 vx 0 vy 隨機取數 10 到 100
    set 飛碟 x to 隨機取數 10 到 150
    角色 飛碟 的類別設為 Enemy
    變數 砲彈 設為 發射物 來自 飛碟 速度 vx 50 vy 75
    角色 砲彈 的類別設為 Enemy
    變數 砲彈 設為 發射物 來自 飛碟 速度 vx -50 vy 75
    角色 砲彈 的類別設為 Enemy
    
```

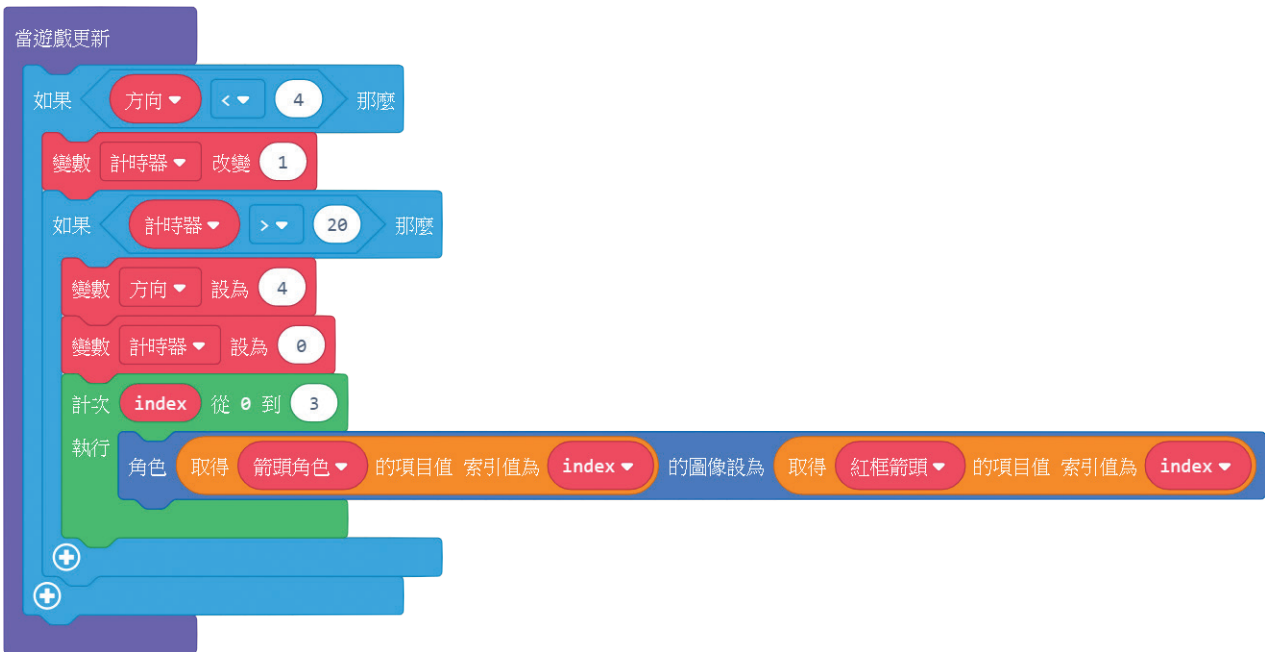
## 第八章

```

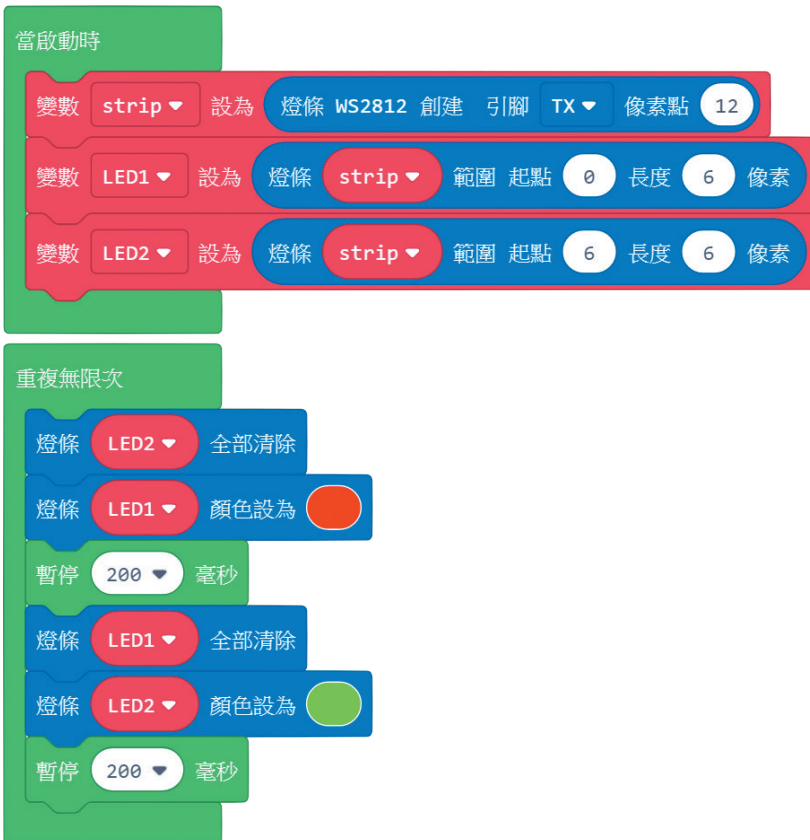
定義函式 排序
    計次 i 從 0 到 5
    執行
        計次 j 從 0 到 5
        執行
            變數 N1 設為 取得 號碼 的項目值 索引值為 i
            變數 N2 設為 取得 號碼 的項目值 索引值為 j
            如果 N1 < N2 那麼
                呼叫 交換 i j
            +
    +

定義函式 交換 num num2
    變數 temp 設為 取得 號碼 的項目值 索引值為 num
    設定 號碼 的項目值 索引值為 num 項目值設為 取得 號碼 的項目值 索引值為 num2
    設定 號碼 的項目值 索引值為 num2 項目值設為 temp
    
```

## 第九章



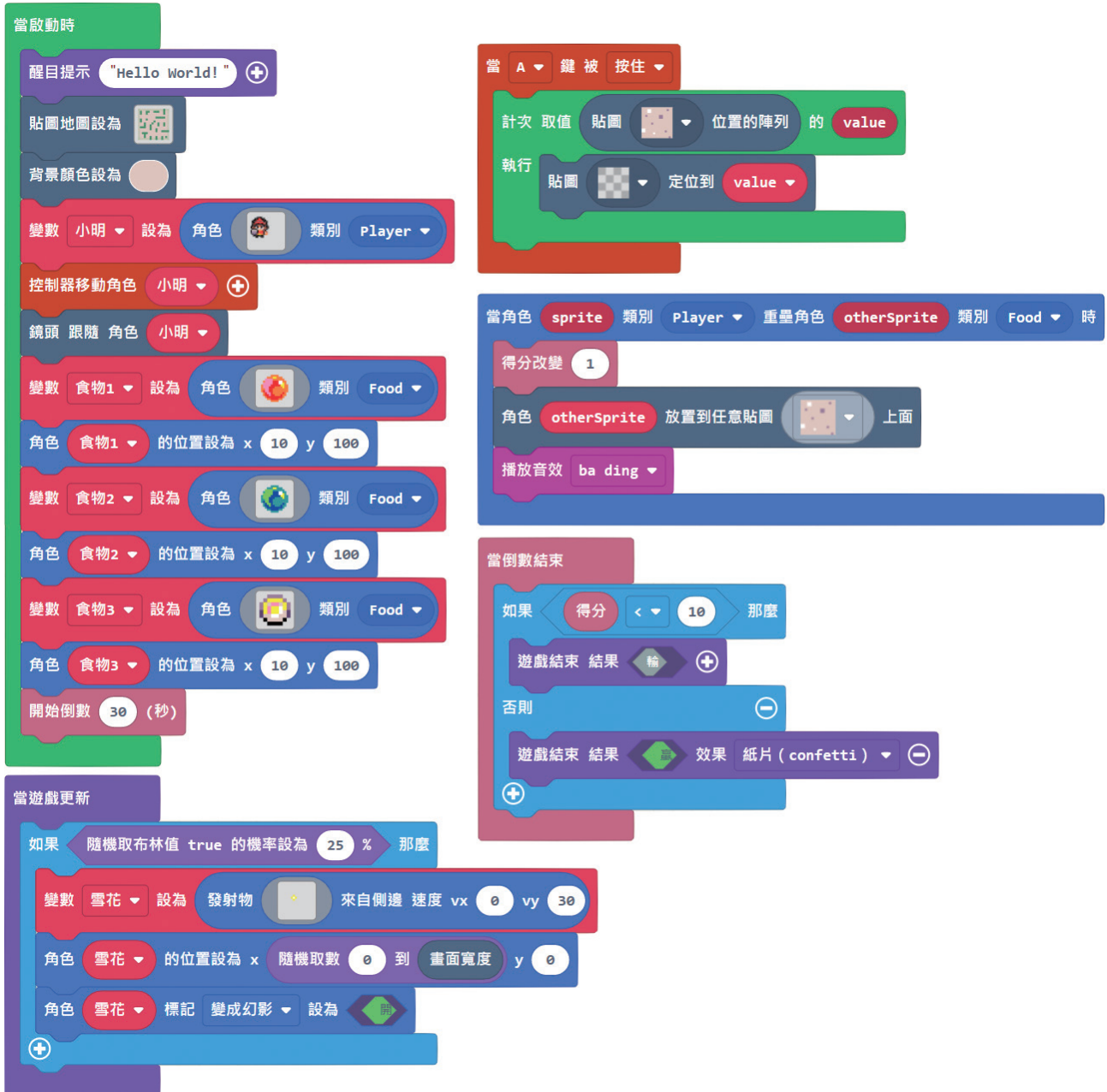
## 第十一章



## 綜合實作題完整程式解答

## 第二章

〔程式檔：ch2 實作題\_迷宮尋寶遊戲〕



The image displays a collection of Scratch code blocks for a maze game, organized into three main sections: '當啟動時' (When the game starts), '當 A 鍵被按住' (When the A key is pressed), and '當遊戲更新' (When the game updates).

**當啟動時 (When the game starts):**

- 醒目標示 "Hello World!"
- 貼圖地圖設為 [maze map]
- 背景顏色設為 [brown]
- 變數 小明 設為 角色 [Player]
- 控制器移動角色 小明
- 鏡頭 跟隨 角色 小明
- 變數 食物1 設為 角色 [Food]
- 角色 食物1 的位置設為 x 10 y 100
- 變數 食物2 設為 角色 [Food]
- 角色 食物2 的位置設為 x 10 y 100
- 變數 食物3 設為 角色 [Food]
- 角色 食物3 的位置設為 x 10 y 100
- 開始倒數 30 (秒)

**當 A 鍵被按住 (When the A key is pressed):**

- 計次 取值 貼圖 [maze map] 位置的陣列 的 value
- 執行 貼圖 [maze map] 定位到 value

**當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Food 時 (When the Player sprite overlaps the Food sprite):**

- 得分改變 1
- 角色 otherSprite 放置到任意貼圖 [maze map] 上面
- 播放音效 ba ding

**當倒數結束 (When the countdown ends):**

- 如果 得分 < 10 那麼
- 遊戲結束 結果 輸
- 否則
- 遊戲結束 結果 效果 紙片 (confetti)

**當遊戲更新 (When the game updates):**

- 如果 隨機取布林值 true 的機率設為 25 % 那麼
- 變數 雪花 設為 發射物 [snowflake] 來自側邊 速度 vx 0 vy 30
- 角色 雪花 的位置設為 x 隨機取數 0 到 畫面寬度 y 0
- 角色 雪花 標記 變成幻影 設為 開

## 第四章

[ 程式檔：ch4 實作題\_射擊隕石遊戲 ]


The image displays several Scratch code blocks for a space shooter game, organized into functional groups:

- 當啟動時 (When the game starts):**
  - 貼圖地圖設為 (Set background to: star field)
  - 效果 星空 (star field) 用於開始畫面 (+)
  - 變數 隕石群 設為 陣列 (Set meteor array to: array)
  - 變數 太空船 設為 角色 (Set spaceship to: sprite)
  - 角色 太空船 標記 留在畫面 設為 (Mark spaceship to stay on screen)
  - 控制器移動角色 太空船 速度 vx 200 vy 200 (-)
  - 生命設為 10
- 當 A 鍵被重複 (When A key is pressed):**
  - 變數 砲彈 設為 發射物 來自 太空船 速度 vx 0 vy -100
- 當角色 sprite 類別 Projectile 重疊角色 otherSprite 類別 Food 時 (When sprite of type Projectile overlaps sprite of type Food):**
  - 角色 otherSprite 摧毀 (+)
  - 得分改變 1
- 當角色 sprite 類別 Food 重疊貼圖 位置 location (When sprite of type Food overlaps image at location):**
  - 生命改變 -1
  - 鏡頭 震動幅度 4 像素 持續 500 毫秒
  - 播放音效 power down
  - 角色 sprite 摧毀 (+)
- 當遊戲更新 (When the game updates):**
  - 如果 得分 > 20 那麼 (If score > 20 then):
    - 變數 延時 設為 500
  - 否則如果 得分 > 10 那麼 (-) (Else if score > 10 then):
    - 變數 延時 設為 800
  - 否則 (-) (Else):
    - 變數 延時 設為 1000
- 重複無限次 (Repeat indefinitely):**
  - 變數 隕石索引值 設為 隨機取數 0 到 隕石群 的長度 - 1
  - 變數 隕石 設為 發射物 取得 隕石群 的項目值 索引值為 隕石索引值 來自側邊 速度 vx 0 vy 隨機取數 10 到 100
  - 角色 隕石 的類別設為 Food
  - set 隕石 x to 隨機取數 10 到 150
  - 暫停 延時 毫秒

## 第十一章

[ 程式檔：ch11 實作題\_小鴨鴨遊戲 ]


**當啟動時**

- 背景顏色設為
- 呼叫 初始化背景
- 變數 小鴨 設為 角色  類別 Player
- 角色 小鴨 的位置設為 x 10 y 100
- 得分設為 0
- 變數 strip 設為 燈條 WS2812 創建 引腳 TX 像素點 12
- 燈條 strip 全部清除
- 變數 flag 設為 0

**重複無限次**

- 如果  地面1 右邊  < 0 那麼
  - set 地面1 左邊 to 地面2 右邊
- 如果  地面2 右邊  < 0 那麼
  - set 地面2 左邊 to 地面1 右邊



**重複無限次**

- 如果  flag = 1 那麼
  - 燈條 strip 顏色設為
  - 暫停 100 毫秒
  - 燈條 strip 全部清除
  - 變數 木頭 設為 發射物  來自側邊 速度 vx 地面1 速度 x (vx) vy 0
  - set 木頭 y to 100
  - 變數 flag 設為 0

**當角色 sprite 類別 Player 重疊角色 otherSprite 類別 Projectile 時**

- 遊戲結束 結果

**定義函式 初始化背景**

- 變數 地面1 設為 角色  類別 Ground
- 變數 地面2 設為 角色  類別 Ground
- 角色 地面1 的位置設為 x 80 y 120
- 角色 地面2 的位置設為 x 80 + 160 y 120
- set 地面1 速度 x (vx) to -100
- set 地面2 速度 x (vx) to -100

**當遊戲更新**

- 得分改變 1
- 如果  小鴨 y ≥ 100 那麼
  - set 小鴨 y to 100

**當遊戲更新 間隔 1000 毫秒**

- change 地面1 速度 x (vx) by -1
- change 地面2 速度 x (vx) by -1
- 如果  隨機取布林值 true 的機率設為 60 % 那麼
  - 變數 flag 設為 1

**當 任何 鍵被 按住**

- 如果  小鴨 y = 100 那麼
  - set 小鴨 速度 y (vy) to -200
  - set 小鴨 加速度 y (ay) to 500

## 附錄二、Arcade 常用積木指令範例


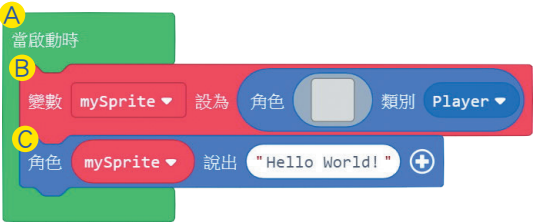
以下範例詳細說明各常用積木指令的使用情境，可與「附錄三、Arcade 積木指令介紹」做對照練習，以下的「圖像」、「函式」、「陣列」、「文字」、「控制台」等積木盒在「進階」積木盒中。






遊戲中關於創建遊戲角色和類別，處理角色的圖像、物理動作、生命週期、效果及角色類別之間的重疊。

角色是存在於屏幕上可移動的個體圖像，每個角色需賦予一個變數名字及類別（可使用預設或自訂類別）。變數名字用於控制角色；類別用於碰撞偵測。

### 範例 1：屏幕顯示文字

屏幕顯示	範例說明
	<p>遊戲中創建一個角色，名稱為「mySprite」，並在屏幕的角色位置上顯示「Hello World!」。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「角色」積木盒中的「變數設為角色」積木指令，創建一個名為「mySprite」的角色，下拉變數清單選項可選「新的變數」或「重新命名變數」（如選重新命名變數，則程式中有相同變數名稱都會一起改變）</p> <p>C. 設定在「mySprite」角色出現的位置顯示“Hello World!”，因為角色貼圖目前預設為透明，所以還看不見，角色的預設位置為屏幕正中央（80,60）</p>


## 範例 2：角色移動自動反彈

螢幕顯示	範例說明
	<p>創建一個角色名稱為「mySprite」，貼圖使用自繪圓圈圖形，在屏幕上角色可自行移動，速度為 vx：50 vy：50，初始位置座標（50,50），碰到屏幕邊緣自動反彈。</p>
積木程式（解答一）	指令解說
	<p>A. 當啟動時執行事件區塊內的程式          B. 使用「角色」積木盒中的「變數設為角色」積木指令，創建一個角色，名稱為「mySprite」          C. 點選空白角色貼圖，使用貼圖編輯器畫圓工具繪製圓圈。（參考 3-2 圖形編輯器）          D. 設定角色中心點初始位置在屏幕座標（100,100）          E. 設定角色等速移動，水平、垂直速度均為正值「50」（往右下移動）          F. 使用「角色」積木盒中的「角色標記 ..」積木指令，設定角色物理動作為「碰到牆反彈」，並將動作設為「開」</p>
積木程式（解答二）	指令解說
	<p>A. 當啟動時執行事件區塊內的程式          B. 使用「角色」積木盒中的「變數設為角色」積木指令，創建一個角色，名稱為「mySprite」          C. 使用「角色」積木盒中的「角色的圖像設為 ..」積木指令，點選空白角色貼圖，使用貼圖編輯器畫圓工具繪製圓圈          D. 使用「角色」積木盒中的「set 設定屬性」積木指令，設定角色中心點距屏幕左側（x）「100」像素          E. 使用「set 設定屬性」積木指令，設定角色中心點距屏幕上面（y）「100」像素          F. 使用「set 設定屬性」積木指令，設定角色等速移動，水平速度（vx）為正值「50」（往右移動）          G. 使用「set 設定屬性」積木指令，設定角色等速移動，垂直速度（vy）為正值「50」（往下移動）          H. 使用「角色」積木盒中的「角色標記 ..」積木指令，設定角色物理動作為「碰到牆反彈」，並將動作設為「開」</p>

## 範例 3：跟隨角色移動

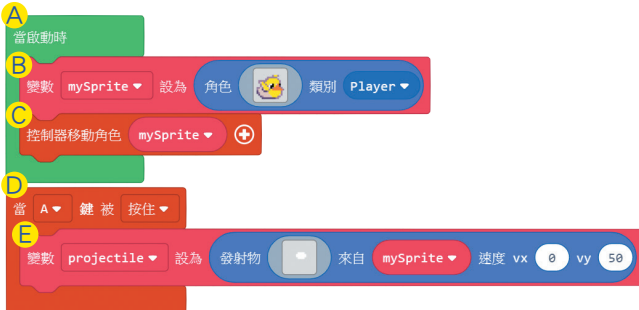
螢幕顯示	範例說明
	<p>續上一範例，再創建第二角色名稱爲「mySprite2」，角色類別爲「Player」，設定跟隨「mySprite」角色，速度 50，並產生噴灑效果。</p>
積木程式	指令解說
	<p>A. 使用「角色」積木盒中的「變數設為角色」積木指令，創建第二個角色，名稱爲「mySprite2」，點選空白角色貼圖，使用貼圖編輯器畫矩形工具繪製一「矩形」</p> <p>B. 使用「角色」積木盒中的「角色跟隨角色」積木指令，設定「mySprite2」跟隨「mySprite」移動，按「+」設定跟隨速度爲「50」</p> <p>C. 使用「角色」積木盒中的「角色開始效果」積木指令，設定「mySprite2」持續產生「噴灑」效果</p>

 控制器 Controller

遊戲中處理控制器按鈕（方向鍵、A、B 鍵）對應於角色的動作，目前支援雙人遊戲控制，點選模擬器下方的  圖示，可查看玩家使用鍵盤時對應的快速鍵。讓遊戲機按鈕對應玩家角色的移動方向及移動速度，當二台遊戲機連線時，可設定多個玩家控制。

## 範例 4：控制器應用及產生發射物

螢幕顯示	範例說明
	<p>創建一個遊戲主角使用鴨子貼圖，可利用方向鍵控制鴨子位置，並且按「A」鍵下蛋。</p>

積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」，點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「鴨子」（duck）圖案</p> <p>C. 設定控制器與角色「mySprite」做連結，能使用方向鍵移動「mySprite」在屏幕中的位置</p> <p>D. 當按「A」鍵時執行事件區塊內的程式</p> <p>E. 使用「角色」積木盒中的「變數設為發射物」積木指令，創建一個發射物，名稱為「projectile」，點選空白發射物貼圖，使用貼圖編輯器繪製白色實心橢圓，發射物設定來自角色「mySprite」，並且向下發射（vx=0，vy=50）</p>

**範例 5：改變（change）屬性應用**

屏幕顯示	範例說明
	<p>創建一個遊戲主角使用鴨子貼圖，可利用左、右方向鍵移動鴨子水平位置</p> <p>※ 直接設定鴨子位置</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」，點選空白角色貼圖，使用貼圖編輯器中的素材庫選擇「鴨子」（duck）圖案</p> <p>C. 當按「左」鍵時執行事件區塊內的程式</p> <p>D. 使用「角色」積木盒中的「change 改變屬性」積木指令，讓角色「mySprite」的水平（x）位置減少「16」像素（往左移動一個貼圖寬度）</p> <p>E. 當按「右」鍵時執行事件區塊內的程式</p> <p>F. 使用「change 改變屬性」積木指令，讓角色「mySprite」的水平（x）位置增加「16」像素（往右移動一個貼圖寬度）</p>

變數 Variables

可以想像變數是一個盒子，每個變數都有一個名字（可使用中文），變數的資料型態是依盒子所放的內容物而決定，可以放數字、文字、貼圖、角色或是一整個清單項目等，變數內容在程式中可以隨時更改，也可以隨時取出使用。

範例 6：變數設定、改變

螢幕顯示	範例說明
	<p>創建一個遊戲主角使用鴨子貼圖，可利用「A」、「B」鍵斜角移動鴨子位置                  ※ 使用變數記錄目前鴨子位置</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式                  B. 創建一個角色，名稱為「mySprite」                  C. 點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「鴨子」（duck）圖案                  D. 使用「變數」積木盒中的「建立一個變數」新增一個「index」變數；再從「變數」積木盒中拖曳出「變數 index 設為 ..」積木指令，並設定變數 index 內容為 0                  E. 當按「A」鍵時執行事件區塊內的程式                  F. 使用「變數」積木盒中的「變數改變」積木指令，將「index」變數內容增加 10                  G. 將角色「mySprite」的水平（x）與垂直（y）位置都設定為「index」變數的內容                  H. 當按「B」鍵時執行事件區塊內的程式                  I. 使用「變數改變」積木指令，將「index」變數內容減少 10                  J. 將角色「mySprite」的水平（x）與垂直（y）位置都設定為「index」變數的內容</p>

 數學 Math

遊戲中關於算術運算、取餘數、隨機取數及四捨五入等數值運算的處理指令。

### 範例 7：隨機取數設定角色位置

屏幕顯示	範例說明
	<p>創建二個遊戲主角並使用漢堡和披薩貼圖代表，利用按「A」鍵往右移動二個角色一段隨機距離（0～15 像素）</p> <p>※ 使用變數 X1、X2 分別記錄二個角色目前的水平位置</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「漢堡」（Burger）圖案</p> <p>D. 設定「mySprite」的初始位置座標為（10,50）</p> <p>E. 創建一個角色，名稱為「mySprite2」</p> <p>F. 點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「披薩」（Pizza）圖案</p> <p>G. 設定「mySprite2」的初始位置座標為（10,100）</p> <p>H. 當按「A」鍵時執行事件區塊內的程式</p> <p>I. 新增一個「X1」變數，設定變數內容為隨機產生 0～15 之間的整數</p> <p>J. 將「mySprite」的水平（x）位置設為「X1」變數的內容</p> <p>K. 新增一個「X2」變數，設定變數內容為隨機產生 0～15 之間的整數</p> <p>L. 將「mySprite2」的水平（x）位置設為「X2」變數的內容</p>

## 文字 Text

針對字串做組合、擷取、分割、轉換相關的文字操作。

### 範例 8：字串組合應用

螢幕顯示	範例說明
	<p>使用字串組合積木指令，在螢幕上顯示“Hello World”的字串訊息。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 新增一個「W1」變數，變數內容為“Hello”（註 1）</p> <p>D. 新增一個「W2」變數，變數內容為“World”</p> <p>E. 使用「角色」積木盒中的「角色說出..」積木指令，讓角色說出由「W1」+ 空白 + 「W2」三者組合而成的文字內容</p> <p><b>註 1</b> 變數內容為文字時需先從「文字」積木盒中取出「空字串」（前後雙引號）指令  放入變數內容中，再輸入文字</p>


## 範例 9：數字轉換成文字型別

螢幕顯示	範例說明
	<p>按「A」鍵在螢幕上顯示 1 ~ 6 的隨機骰子點數。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式          B. 創建一個角色，名稱為「mySprite」          C. 當按「A」鍵時執行事件區塊內的程式          D. 使用「變數」積木盒，建立一個變數命名為「R」，變數內容為隨機產生 0 ~ 5 的數後再加 1（註 2）          E. 使用「角色」積木盒中的「角色說出..」積木指令，讓角色說出變數「R」的內容（註 3）</p> <p><b>註 2</b> 注意順序，先取隨機數後再加 1  <b>註 3</b> 因變數「R」的資料型態是數值不是文字，必須使用「文字」積木盒中的「轉換成文字型別」積木，才能將數值型態變數轉成文字型態變數，「R」的內容也才能顯示出來</p>

**邏輯 Logic**

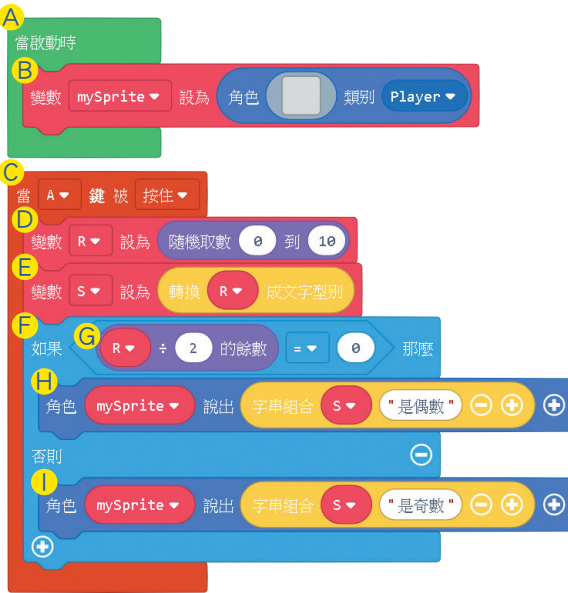
遊戲中關於判斷、比較、邏輯運算等處理指令，邏輯運算或比較的結果只有兩種，不是「true」（真），就是「false」（假），即所謂的布林值。

**範例 10：依機率做為執行條件**

螢幕顯示	範例說明
	<p>按「A」鍵在螢幕中隨機產生一個布林值，但只有 1/4（25%）的機會是「true」，其餘為「false」。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式              B. 創建一個角色，名稱為「mySprite」              C. 設定在「mySprite」角色出現的位置顯示“按 A 鍵隨機產生”文字              D. 當按「A」鍵時執行事件區塊內的程式              E. 使用「邏輯」積木盒中的「如果..那麼..否則..」積木指令              F. 條件設定使用「數學」積木盒中的「隨機..機率設為..」積木指令，並將機率設為「25%」（註 4）              G. 使用「角色」積木盒中的「角色說出..」積木指令，讓角色說出“結果為 true”文字              H. 讓角色說出“結果為 false”文字</p> <p><b>註 4</b> 如果條件成立（true），則執行「那麼」下面程式區塊（G）；如果條件不成立（false），則執行「否則」下面程式區塊（H）</p>

**範例 11：依餘數做為執行條件**

螢幕顯示	範例說明
	<p>按「A」鍵在螢幕上隨機產生一個 0～10 的數字，並能顯示該數是奇數或是偶數。</p>


積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 當按「A」鍵時執行事件區塊內的程式</p> <p>D. 使用「變數」積木盒，建立一個變數命名為「R」，變數內容為隨機產生 0～10 的數值</p> <p>E. 建立一個變數「S」，變數內容為將整數變數「R」的數值轉換成文字型態，以方便讓角色說出內容</p> <p>F. 使用「邏輯」積木盒中的「如果 .. 那麼 .. 否則 ..」積木指令</p> <p>G. 條件設定為「當 R 除以 2 的餘數等於 0」時，分別使用「邏輯」積木盒中的「等於」積木指令和「數學」積木盒中的「餘數」積木指令（註 5）</p> <p>H. 當上述條件成立，表示「R」的值為偶數，故屏幕顯示“xx 是偶數”（註 6）</p> <p>I. 當上述條件不成立，表示「R」的值為奇數，故屏幕顯示“xx 是奇數”（註 6）</p> <p><b>註 5</b> 能被 2 所整除的數即為偶數，否則為奇數</p> <p><b>註 6</b> 「xx」為已將 R 轉成文字型態的字串 S 的內容</p>

**● 遊戲 Game**

遊戲中關於遊戲運行、遊戲結果的控制和提示文字、詢問玩家等訊息的處理指令，遊戲過程中的時間控制、提示文字和顯示遊戲結果。

**範例 12：醒目提示訊息應用**

屏幕顯示	範例說明
	<p>在屏幕上顯示“按 A 開始”的字串提示訊息，並等待使用者按「A」繼續執行程式。</p>

積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「遊戲」積木盒中的「醒目提示」積木指令，在屏幕上顯示“按 A 開始”</p>

**範例 13：對話方塊應用及遊戲更新事件**

屏幕顯示	範例說明
	<p>在屏幕上創建一個角色後，在屏幕下方顯示對話方塊，提示“按 A 後開始計時”，並且將對話方塊的游標改成主角圖示，當使用者按「A」鍵，屏幕顯示開始計時。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「英雄」（hero）圖案</p> <p>D. 新增一個變數「Counter」，變數內容設為 0</p> <p>E. 使用「遊戲」積木盒中的「對話視窗的文字顏色設為 ..」積木指令，並設為「紅色」</p> <p>F. 使用「遊戲」積木盒中的「對話視窗的游標設為 ..」積木指令，並設為「英雄」圖示</p> <p>G. 使用「遊戲」積木盒中的「對話內容」積木指令，並設為“按 A 後開始計時”，訊息對話內容顯示於屏幕「下」方</p> <p>H. 當遊戲執行時每隔 1000 毫秒（1 秒），執行事件區塊內的程式</p> <p>I. 變數「Counter」內容值加 1（註 7）</p> <p>J. 使用「角色」積木盒中的「角色說出 ..」積木指令，讓角色說出文字內容，由於變數「Counter」的資料型態是數值不是文字，需先將數值轉換成文字型別</p> <p><b>註 7</b> 變數「Counter」內容值每隔一秒將加 1</p>

## 範例 14：角色碰撞事件

屏幕顯示	範例說明
	<p>在屏幕下方創建一個「小鴨」角色，可使用方向鍵控制「小鴨」位置，「小鴨」正上方每隔 0.5 秒會掉下一顆「草莓」，如果「小鴨」被「草莓」碰到則遊戲結束。</p>
積木程式	指令解說
	<ol style="list-style-type: none"> <li>A. 當啟動時執行事件區塊內的程式</li> <li>B. 創建一個角色，下拉名稱清單，選「重新命名變數 ..」將變數名稱更改為「小鴨」</li> <li>C. 點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「鴨子」（duck）圖案</li> <li>D. 設定控制器與角色「小鴨」做連結，能使用方向鍵移動「小鴨」在屏幕中的位置</li> <li>E. 當遊戲執行時每隔 500 毫秒（0.5 秒），執行事件區塊內的程式</li> <li>F. 使用「角色」積木盒中的「變數設為發射物來自側邊」積木指令，更改發射物名稱為「草莓」（註 8）</li> <li>G. 點選空白發射物貼圖，使用貼圖編輯器中的「素材庫」選擇「草莓」（Strawberry）圖案，將水平速度（vx）改為「0」</li> <li>H. 使用「角色」積木盒中的「set 設定屬性」積木指令，角色選擇「草莓」</li> <li>I. 使用「角色」積木盒中的「獲取角色 x 屬性」積木指令，讓「草莓」的水平位置設定為「小鴨」目前的水平位置</li> <li>J. 使用「角色」積木盒中的「角色類別重疊角色類別」積木事件，設定為當「Player」類別重疊「Projectile」類別時執行事件區塊內的程式</li> <li>K. 使用「遊戲」積木盒中的「遊戲結束」積木事件，設定結果為「輸」</li> </ol> <p><b>註 8</b> 發射物類別預設為「Projectile」</p>

音效 Music


遊戲中關於製作音樂、演奏旋律和音效等處理指令，可設定音量、演奏速度等。

範例 15：演奏旋律及播放音效

螢幕顯示	範例說明
	<p>程式開始時先播放「do、re、mi、fa、so、la、si、do」八個音階，音量設為「20%」。</p> <p>當按「A」鍵時，循環播放「ba ding」音效，一直等到按「B」鍵時才停止播放。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「音效」積木盒中的「演奏旋律」積木指令，點選空白旋律圖示後進入「素材庫」，選擇「Reverse」旋律</p> <p>C. 使用「音效」積木盒中的「音量設為..」積木指令，將音量設為「20」</p> <p>D. 當按「A」鍵時執行事件區塊內的程式</p> <p>E. 使用「音效」積木盒中的「循環播放音效」積木指令，音效選項下拉選單選擇「ba ding」</p> <p>F. 當按「B」鍵時執行事件區塊內的程式</p> <p>G. 使用「音效」積木盒中的「停止播放音效」積木指令，結束循環播放「ba ding」音效</p>

範例 16：播放「小星星」旋律

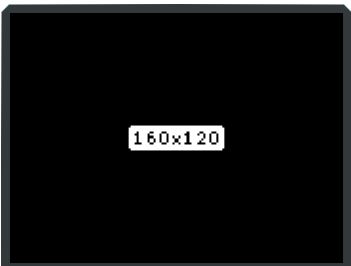
螢幕顯示	範例說明
	<p>當按「A」鍵時播放「小星星」的旋律一次。</p> <p style="text-align: center;">小星星簡譜</p> <p style="text-align: center;">1155665 4433221 5544332 5544332 1155665 4433221</p>

積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 使用「角色」積木盒中的「角色說出..」積木指令，在角色位置上顯示“按 A 演奏小星星”</p> <p>D. 當按「A」鍵時執行事件區塊內的程式</p> <p>E. 使用「音效」積木盒中的「演奏旋律」積木指令，點選空白旋律圖示後進入「編輯器」，由左至右分別點選音階「1155665」</p> <p>F. 同上，並選擇音階「4433221」</p> <p>G. 同上，並選擇音階「5544332」</p> <p>H. 複製上一個旋律積木指令</p> <p>I. 複製第一個旋律的積木指令</p> <p>J. 複製第二個旋律的積木指令</p>

 場景 Scene

遊戲中關於屏幕背景顏色設定、貼圖地圖製作及玩家視角設置、處理貼圖 (Tiles) 和角色之間作用的處理指令。

範例 17：顯示系統屏幕解析度

屏幕顯示	範例說明
	<p>在屏幕上顯示由系統讀出之屏幕解析度（寬 × 高）。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 新增一個變數並改名為「寬度」</p> <p>D. 使用「場景」積木盒中的「畫面寬度」積木指令，將所獲得的數值轉換為文字型態</p> <p>E. 新增一個變數並改名為「高度」</p> <p>F. 使用「場景」積木盒中的「畫面高度」積木指令，將所獲得的數值轉換為文字型態</p> <p>G. 使用「角色」積木盒中的「角色說出..」積木指令</p> <p>H. 使用「文字」積木盒中的「字串組合」積木指令，顯示「寬度」變數內容、「x」文字、「高度」變數內容三者組合出的字串</p>

## 範例 18：角色移動至任意貼圖上面

屏幕顯示	範例說明
	<p>在 32×32 的貼圖地圖上製作背景，地圖上放置一些「森林」貼圖，並且讓屏幕產生紙片紛飛的效果。</p> <p>再創建一個「漢堡」角色，當按下「A」鍵時，「漢堡」會隨機出現在任一個「森林」貼圖的位置上，並在「漢堡」下方顯示目前座標，鏡頭必須隨「漢堡」移動，否則「漢堡」有可能會出現在屏幕外。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「場景」積木盒中的「貼圖地圖」積木指令</p> <p>C. 點選空白地圖，使用地圖編輯器，隨機放置一些「forest」（森林）貼圖（註 9）</p> <p>D. 使用「場景」積木盒中的「效果用於開始畫面」積木指令</p> <p>E. 創建一個角色，名稱為「mySprite」</p> <p>F. 點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「漢堡」（Burger）圖案</p> <p>G. 使用「角色」積木盒中的「角色標記」積木指令，下拉標記選單選擇「顯示座標與速度」，並將動作設為「開」</p> <p>H. 使用「場景」積木盒中的「鏡頭跟隨角色」積木指令，設定屏幕跟隨角色「mySprite」</p> <p>I. 當按「A」鍵時執行事件區塊內的程式</p> <p>J. 使用「場景」積木盒中的「角色放置到任意貼圖」積木指令，下拉貼圖選單選擇「forest」（森林）貼圖</p> <p><b>註 9</b> 可參考章節 2-2 遊戲背景製作</p>

## 範例 19：角色碰撞指定貼圖事件

屏幕顯示	範例說明
	<p>新增一張 32×32 的空白貼圖地圖，地圖上第 2 欄第 3 列由程式產生一個「森林」貼圖，再創建一個「漢堡」角色，可使用方向鍵移動。當「漢堡」角色移動時碰到「森林」貼圖，「漢堡」角色顯示“撞到樹了”文字 0.5 秒後消失。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「場景」積木盒中的「貼圖地圖」積木指令</p> <p>C. 使用「場景」積木盒中的「貼圖定位到 ..」積木指令，下拉貼圖選單選擇「forest」（森林）貼圖，將貼圖定位到第 2 欄第 3 列（註 10）</p> <p>D. 創建一個角色，名稱為「mySprite」，點選空白角色貼圖，使用貼圖編輯器中的素材庫選擇「漢堡」（Burger）圖案</p> <p>E. 設定控制器與角色「mySprite」做連結，能使用方向鍵移動「mySprite」在屏幕中的位置</p> <p>F. 使用「場景」積木盒中的「角色類別重疊貼圖」積木事件</p> <p>G. 設定當「Player」重疊「forest」（森林）貼圖時，執行事件區塊內的程式</p> <p>H. 使用「角色」積木盒中的「角色說出 ..」積木指令，讓「mySprite」角色說出“撞到樹了”的文字內容，按「+」設定「500」毫秒（0.5 秒）後文字消失</p> <p><b>註 10</b> 下拉貼圖選單中如無「森林」貼圖，必須先到貼圖地圖（點空白貼圖地圖）中點選一次「森林」貼圖，才會出現在下拉選單中。</p>

## 範例 20：貼圖定位到貼圖地圖

螢幕顯示	範例說明
	<p>新增一張 32×32 的空白貼圖地圖，地圖上第 2 欄第 3 列產生一個「森林」貼圖，並將其設定為「牆壁」，再創建一個「漢堡」角色，可使用方向鍵移動。</p> <p>當「漢堡」角色移動時碰到「牆壁」貼圖，播放「ba ding」音效。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「場景」積木盒中的「貼圖地圖」積木指令</p> <p>C. 使用「場景」積木盒中的「貼圖定位到」積木指令，下拉選單選擇「forest」（森林）貼圖，將貼圖定位到第 2 欄第 3 列（註 11）</p> <p>D. 使用「場景」積木盒中的「牆壁定位到」積木指令，動作設為「開」，並將牆壁定位到第 2 欄第 3 列（註 12）</p> <p>E. 創建一個角色，名稱為「mySprite」，點選空白角色貼圖，使用貼圖編輯器中的素材庫選擇「漢堡」（Burger）圖案</p> <p>F. 設定控制器與角色「mySprite」做連結，能使用方向鍵移動「mySprite」在屏幕中的位置</p> <p>G. 使用「場景」積木盒中的「類別碰到牆壁」積木事件，設定當角色類別「Player」碰到「牆壁」時，執行事件區塊內的程式</p> <p>H. 使用「音效」積木盒中的「播放音效」積木指令，播放「ba ding」音效</p> <p><b>註 11</b> 如果下拉選單沒有所需的貼圖，請參考上一範例註 10</p> <p><b>註 12</b> 設為牆壁的位置角色將無法進入（除非角色設為幻影）</p>

 資訊 Info

遊戲中關於改變遊戲得分或生命值、遊戲倒數計時與生命狀態偵測等指令。

### 範例 21：生命值、得分值及遊戲倒數

螢幕顯示	範例說明
	<p>程式開始時設定倒數「10 秒鐘」、生命值設為「3」、得分設為「0」。</p> <p>遊戲進行中得分值持續加 1 分，當按「A」鍵，生命值減 1。</p> <p>當倒數結束或生命值为 0 時，結束遊戲並顯示得分。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 使用「資訊」積木盒中的「開始倒數」積木指令，並將倒數時間設為「10」秒</p> <p>C. 使用「生命設為 ..」積木指令，並將生命值設為「3」</p> <p>D. 使用「得分設為 ..」積木指令，並將得分值設為「0」</p> <p>E. 使用「遊戲」積木盒中的「當遊戲更新」積木事件，程式執行中會不斷觸發此事件，並重複執行事件區塊內的程式</p> <p>F. 使用「資訊」積木盒中的「得分改變」積木指令，將得分值加 1</p> <p>G. 當按「A」鍵時，執行事件區塊內的程式</p> <p>H. 使用「資訊」積木盒中的「生命改變」積木指令，將生命值減 1</p>

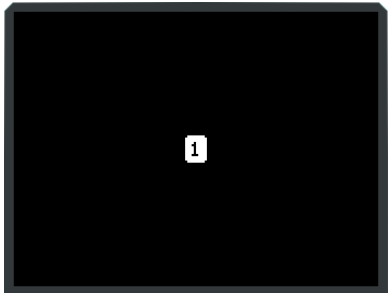
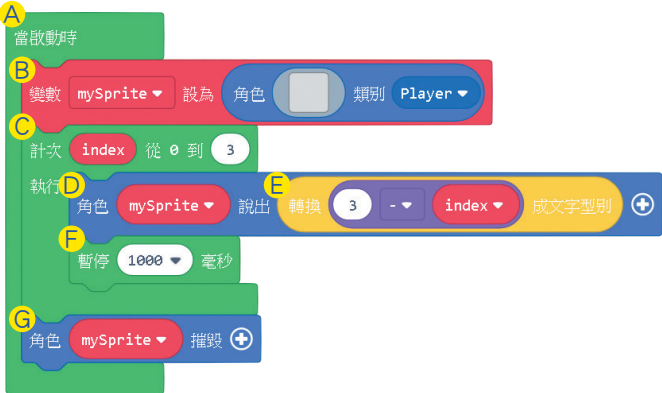
 迴圈 Loops

遊戲中關於重複執行相同動作、依條件重複執行或依清單列舉重複執行等指令，也可用於列舉陣列中的項目或畫面中的所有相同類別的角色。

## 範例 22：固定迴圈應用

螢幕顯示	範例說明
	<p>程式開始時創建一個「mySprite」角色，選擇使用「英雄」圖示，每隔 0.5 秒角色往右下移動 10 像素，並發出「pew pew」音效，連續 5 次。</p>
積木程式	指令解說
	<ol style="list-style-type: none"> <li>A. 當啟動時執行事件區塊內的程式</li> <li>B. 創建一個角色，名稱為「mySprite」，點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「英雄」（hero）圖案</li> <li>C. 使用「迴圈」積木盒中的「重複」積木指令，設定重複執行「5」次區塊內的程式</li> <li>D. 使用「音效」積木盒中的「播放音效」積木指令，播放「pew pew」音效</li> <li>E. 使用「角色」積木盒中的「change 改變屬性」積木指令，讓角色「mySprite」的水平（x）位置增加「10」像素</li> <li>F. 使用「角色」積木盒中的「change 改變屬性」積木指令，讓角色「mySprite」的垂直（y）位置增加「10」像素</li> <li>G. 使用「遊戲」積木盒中的「暫停」積木指令，設定程式暫停 500 毫秒（0.5 秒）再往下繼續執行</li> </ol>

## 範例 23：計次迴圈應用

螢幕顯示	範例說明
	<p>程式開始時創建一個「mySprite」角色，在螢幕上顯示「3、2、1、0」自動倒數計時，計時完後將角色刪除。</p>
積木程式	指令解說
	<ol style="list-style-type: none"> <li>A. 當啟動時執行事件區塊內的程式</li> <li>B. 創建一個角色，名稱為「mySprite」</li> <li>C. 使用「迴圈」積木盒中的「計次」積木指令，設定迴圈結束值為 3，共執行 4 次迴圈區塊內的程式（註 13）</li> <li>D. 使用「角色」積木盒中的「角色說出..」積木指令</li> <li>E. 將「3 - index」的運算結果轉換成文字型態，顯示在角色位置上</li> <li>F. 使用「遊戲」積木盒中的「暫停」積木指令，設定程式暫停 1000 毫秒（1 秒）再往下繼續執行</li> <li>G. 迴圈結束後，使用「角色」積木盒中的「角色摧毀」積木指令，刪除角色「mySprite」</li> </ol> <p><b>註 13</b> 變數「index」內容從 0 開始，每次自動增加 1</p>



遊戲中關於創建圖像、填充、繪製和像素設置等與點陣圖相關的處理指令，可設定圖像翻轉或顏色取代。

### 範例 24：繪製矩形圖像

屏幕顯示	範例說明
	<p>在屏幕上產生一個填滿黃色的矩形圖形（寬、高為屏幕的一半），並且有紅色邊框。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 新增一個變數並改名為「W」，變數內容為屏幕寬度的一半（畫面寬度 ÷ 2）</p> <p>C. 新增一個變數並改名為「H」，變數內容為屏幕高度的一半（畫面高度 ÷ 2）</p> <p>D. 新增一個變數並改名為「PIC」</p> <p>E. 使用「圖像」積木盒中的「圖像創建」積木指令，從「變數」積木盒中拖曳出「W」放在寬度屬性中；拖曳出「H」放在高度屬性中</p> <p>F. 使用「圖像」積木盒中的「顏色填滿」積木指令，選擇「黃色」填滿「PIC」圖像</p> <p>G. 使用「圖像」積木盒中的「繪製矩形」積木指令，從「變數」積木盒中拖曳出「W」放在寬度屬性中；拖曳出「H」放在高度屬性中，再選擇「紅色」</p> <p>H. 創建一個角色，名稱為「mySprite」，從「變數」積木盒中拖曳出「PIC」，將角色圖示改為「PIC」圖像（註 14）</p> <p><b>註 14</b> 一定要將圖像指定給角色，圖像才能顯示在屏幕上</p>

## 範例 25：填滿矩形圖像

螢幕顯示	範例說明
	<p>在螢幕上畫出 Arcade 的 16 種顏色的色塊，每個色塊 10×50 像素（第 1 種顏色為透明，最後的顏色為黑色）。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 新增一個變數並改名為「picture」，變數內容使用「圖像」積木盒中的「圖像創建」積木指令，寬度設為「160」；高度設為「50」</p> <p>C. 創建一個角色，名稱為「mySprite」，從「變數」積木盒中拖曳出「picture」放在角色圖示中</p> <p>D. 使用「迴圈」積木盒中的「計次」積木指令，迴圈變數名稱更改為「i」，設定迴圈結束值為 15，變數「i」內容從 0 開始，每次自動增加 1，共執行 16 次迴圈區塊內的程式</p> <p>E. 設定變數「R」的內容為迴圈次數 (i) * 10（註 15）</p> <p>F. 使用「圖像」積木盒中的「填滿矩形」積木指令，從「變數」積木盒中拖曳出「R」放在座標 x 屬性中；拖曳出「i」放在顏色屬性中（註 16）</p> <p>G. 畫白色外框線，使用「圖像」積木盒中的「繪製矩形」積木指令，設定寬度為「160」；高度為「50」；顏色為「白色」</p> <p><b>註 15</b> 變數「R」內容為每個色塊的水平 (x) 開始位置</p> <p><b>註 16</b> Arcade 的顏色設定可直接使用顏色編號 (0 ~ 15)</p>

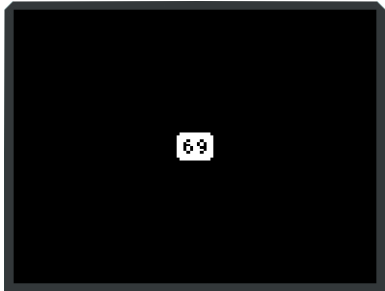
範例 26：圖像水平、垂直翻轉

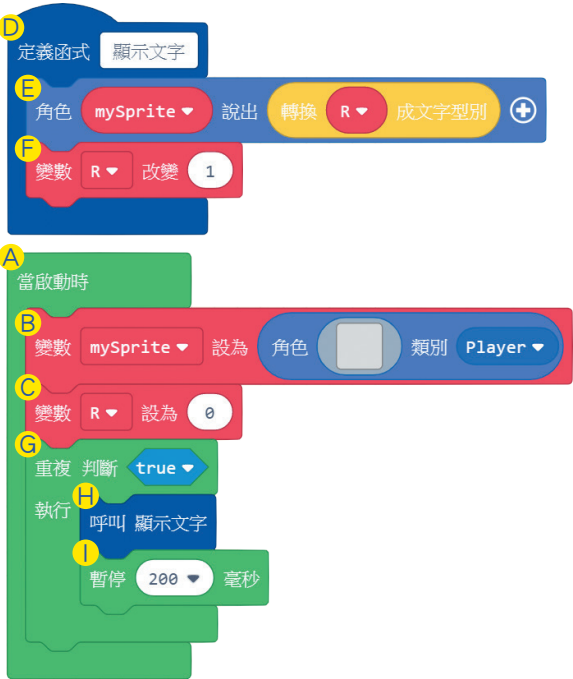
螢幕顯示	範例說明
	<p>程式執行時在螢幕顯示「鴨子」圖像，當按「A」鍵時「鴨子」圖像直向翻轉，當按「B」鍵時「鴨子」圖像橫向翻轉。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式              B. 新增一個變數並改名為「鴨子圖案」              C. 變數內容使用「圖像」積木盒中的「一個圖像」積木指令，並下拉圖像清單選擇「鴨子」(duck) 圖案              D. 創建一個角色，名稱為「小鴨」              E. 使用「變數」積木盒中拖曳出「鴨子圖案」變數放入空白角色貼圖中              F. 當按「A」鍵時，執行事件區塊內的程式              G. 使用「圖像」積木盒中的「圖像直向翻轉」積木指令              H. 當按「B」鍵時，執行事件區塊內的程式              I. 使用「圖像」積木盒中的「圖像橫向翻轉」積木指令</p>

$f(x)$  函式 Functions

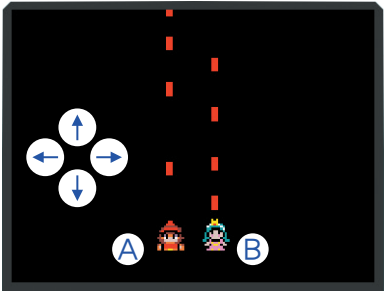
創建一個能夠重複利用的片段程式放在自訂函式中，當在程式中需要這段程式時，只要呼叫該函式即可，使用函式可以很容易地追蹤錯誤，也能增加程式碼的可閱讀性。

範例 27：自訂函式應用

螢幕顯示	範例說明
	<p>創建一個「顯示文字」的自訂函式，功能為將變數「R」內容加 1，並將變數「R」數值內容顯示在螢幕上。              程式開始後每隔 0.2 秒呼叫一次函式。</p>

積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱為「mySprite」</p> <p>C. 使用「變數」積木盒中的「變數設為 ..」積木指令，建立一個變數命名為「R」</p> <p>D. 使用「函式」積木盒中的「建立一個函式 ..」積木，函式名稱改為「顯示文字」，未來在程式中任何位置「呼叫 顯示文字」指令，就會執行函式區塊內的程式</p> <p>E. 使用「角色」積木盒中的「角色說出 ..」積木指令，讓角色說出由變數「R」的內容轉換為文字型別的文字內容</p> <p>F. 變數「R」內容加 1</p> <p>G. 使用「迴圈」積木盒中的「重複 判斷」積木指令，此指令判斷條件如果成立會繼續執行迴圈區塊內的程式。因為判斷條件設為「true」表示會無限次重複執行迴圈區塊內的程式</p> <p>H. 使用「函式」積木盒中的「呼叫 顯示文字」積木指令，程式會去執行「顯示文字」函式中的程式，執行完回來後再往下執行</p> <p>I. 使用「遊戲」積木盒中的「暫停」積木指令，設定程式暫停 200 毫秒（0.2 秒）再往下繼續執行，因為後面沒有程式了，所以會再一次判斷條件，決定是否繼續執行迴圈內的程式</p>

範例 28：自訂傳參數函式應用

屏幕顯示	範例說明
	<p>創建兩個角色，分別叫小明和小美，小明是使用方向鍵來控制移動，小美則是固定在小明的右方 20 像素。</p> <p>創建一個「發射子彈」的自訂函式，供兩個角色呼叫使用。</p> <p>當按「A」鍵，小明發射子彈。</p> <p>當按「B」鍵，小美發射子彈。</p>

積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個角色，名稱改為「小明」，點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「英雄」（hero）圖案</p> <p>C. 設定控制器與角色「小明」做連結，能使用方向鍵移動「小明」在屏幕中的位置</p> <p>D. 創建一個角色，名稱改為「小美」，點選空白角色貼圖，使用貼圖編輯器中的「素材庫」選擇「公主」（princess）圖案</p> <p>E. 使用「遊戲」積木盒中的「當遊戲更新」積木事件，程式執行中會不斷觸發此事件，並重複執行事件區塊內的程式</p> <p>F. 使用「角色」積木盒中的「角色的位置設為 ..」積木指令，將「小美」的水平（x）位置設為「小明」目前的水平（x）位置 +20 像素，垂直（y）位置與小明一樣</p> <p>G. 使用「函式」積木盒中的「建立一個函式 ..」積木，函式名稱改為「發射子彈」，並新增一個「Sprite」（角色）參數，未來在程式中任何位置呼叫「發射子彈」函式時，需帶入一個「角色」當來源參數</p> <p>H. 使用「角色」積木盒中的「變數設為發射物來自 ..」積木指令；更改發射物名稱為「子彈」          點選空白發射物貼圖，使用貼圖編輯器繪製子彈外型          水平速度（vx）設為「0」；垂直速度（vy）設為「-100」（向上移動）          拖曳「發射子彈」函式積木上的「mySprite」參數至發射物的來自角色屬性上</p> <p>I. 當按「A」鍵時執行事件區塊內的程式</p> <p>J. 使用「函式」積木盒中的「呼叫 發射子彈」積木指令，下拉參數選單選擇「小明」，當做函式中發射子彈的來源角色</p> <p>K. 當按「B」鍵時執行事件區塊內的程式</p> <p>L. 使用「呼叫 發射子彈」積木指令，下拉參數選單選擇「小美」，當做函式中發射子彈的來源角色</p>



## ☰ 陣列 Arrays

陣列是一個清單，陣列變數指的是一組相同名字的清單，每個項目值使用索引編號來指定，清單中可以放數字、文字、圖像或角色等，並且可以在陣列中的任何位置添加或刪除清單中的項目。

### 範例 29：陣列變數應用

螢幕顯示	範例說明
	<p>創建一個有三種水果圖示的陣列變數，分別為水果(0)、水果(1)、水果(2)。每隔 0.5 秒從屏幕左上角隨機出現一種水果，往右下角移動。</p>
積木程式	指令解說
	<ol style="list-style-type: none"> <li>當啟動時執行事件區塊內的程式</li> <li>使用「陣列」積木盒中的「變數設為陣列」積木指令（數字或文字陣列都可以），將變數名稱改為「水果」</li> <li>按「+」新增一個空白項目，陣列清單的三個項目內容，分別放置「圖像」積木盒中的「一個圖像」（漢堡圖示）積木指令，並下拉圖像選單，依序選擇三種不同的水果圖示</li> <li>當遊戲執行時每隔 500 毫秒（0.5 秒），執行事件區塊內的程式</li> <li>使用「變數」積木盒新增一個變數，變數名稱設為「R」</li> <li>使用「陣列」積木盒中的「取得 .. 的項目值」積木指令，取得來源為「水果」陣列變數，索引值使用「數學」積木盒中的「隨機取數」積木指令，設定 0 到 2（水果 0 ~ 水果 2）</li> <li>使用「角色」積木盒中的「變數設為發射物來自側邊」積木指令，更改發射物名稱為「掉落物」；下拉發射物清單選擇「R」；水平速度（vx）、垂直速度（vy）設為「100」（向右下移動）</li> </ol>

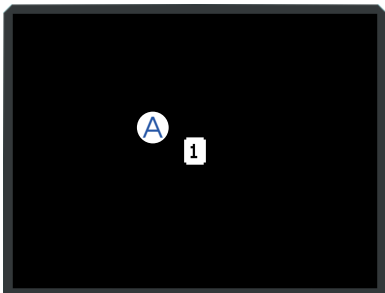
### 範例 30：字串分割應用

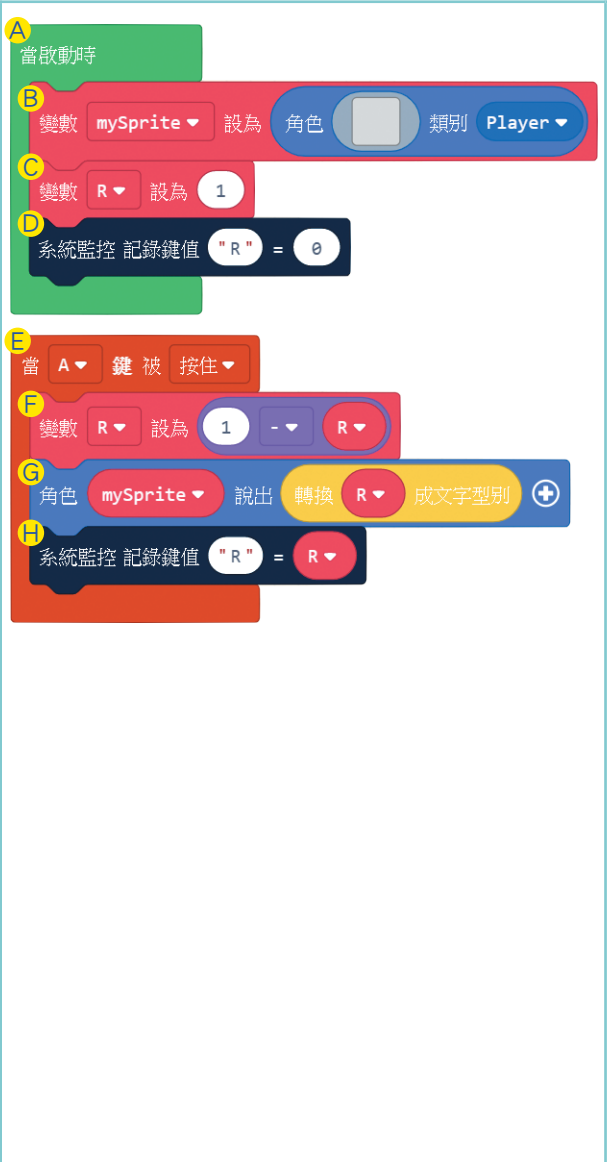
螢幕顯示	範例說明
	<p>開始執行程式後，按「A」鍵，屏幕上每隔 0.5 秒依序顯示「甲」、「乙」、「丙」、「丁」、「戊」、「己」、「庚」、「辛」、「壬」、「癸」，由於中文編碼問題，部分中文字可能無法顯示。</p>
積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式              B. 創建一個名為「mySprite」的角色              C. 新增一個名為「VAR」的變數，內容值為「甲、乙、丙、丁、戊、己、庚、辛、壬、癸」（使用「、」分隔）              D. 在角色的位置上顯示“按 A 鍵顯示十天干”文字              E. 當按「A」鍵時執行事件區塊內的程式              F. 新增一個名為「ARR」的變數，內容值為使用「文字」積木盒中的「字串拆分」積木指令，將變數「VAR」的文字內容依「、」做為拆子字串的分隔符號，執行後結果為：ARR (0) = “甲”、ARR (1) = “乙” … ARR (9) = “癸”              G. 使用「迴圈」積木盒中的「計次取值」積木指令，依「ARR」陣列清單數量，依次取出項目內容放入「value」變數中              H. 使用「角色」積木盒中的「角色說出..」積木指令，讓角色說出「value」變數文字內容              I. 使用「遊戲」積木盒中的「暫停」積木指令，設定程式暫停 500 毫秒（0.5 秒）再往下繼續執行</p>

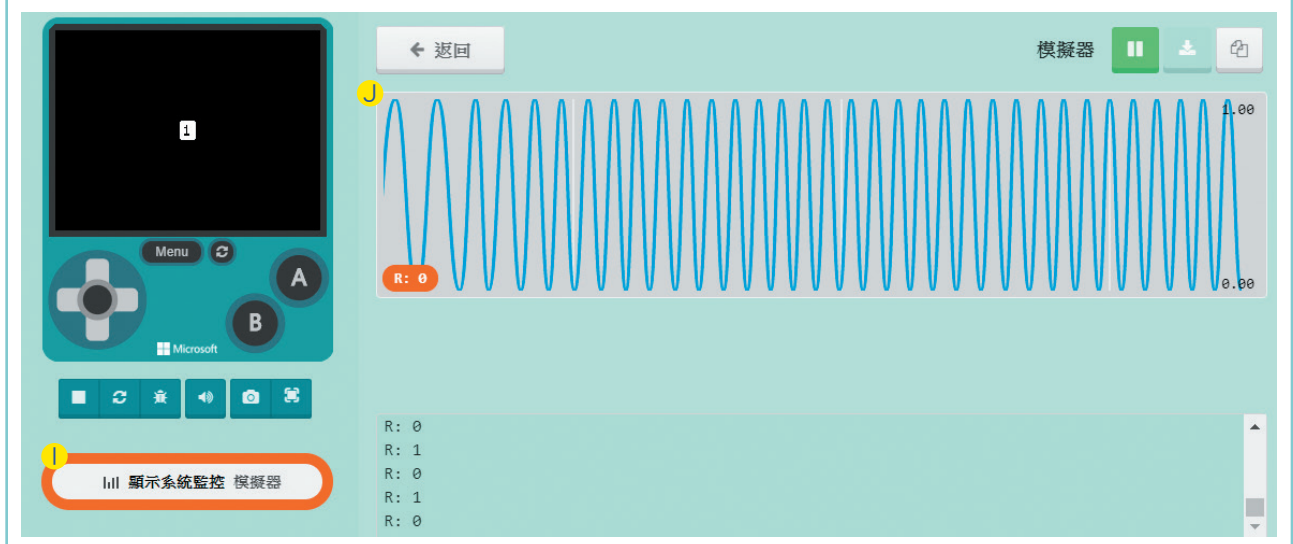


監控遊戲機執行時的一些記錄值，有助於除錯或了解遊戲中未顯示的細節。

### 範例 31：系統監控應用

螢幕顯示	範例說明
	<p>開始執行程式後，持續點按「A」鍵，屏幕上「0」、「1」交互顯示，並將值傳送至「系統監控器」，並以圖形和文字顯示接收值。</p>

積木程式	指令解說
	<p>A. 當啟動時執行事件區塊內的程式</p> <p>B. 創建一個名為「mySprite」的角色</p> <p>C. 新增一個名為「R」的變數，變數內容設為「1」</p> <p>D. 使用「控制台」積木盒中的「系統監控」積木指令，讓模擬器下方顯示「系統監控」圖示，記錄值先送出“R” = 0</p> <p>E. 當按「A」鍵時執行事件區塊內的程式</p> <p>F. 使用「變數」積木盒中的「變數 .. 設為」積木指令，變數名稱下拉選單選擇「R」，變數內容為「1-R」（註 17）</p> <p>G. 使用「角色」積木盒中的「角色說出 ..」積木指令，讓角色說出由數值變數「R」轉換為文字型別的文字內容</p> <p>H. 使用「控制台」積木盒中的「系統監控」積木指令，將文字“R”和變數「R」的數值內容傳送給系統監控記錄</p> <p>I. 點選模擬器下方的「系統監控」圖示，開啟系統監控器</p> <p>J. 系統監控器會持續顯示程式中送到系統監控的記錄值，並以圖形和文字顯示</p> <p><b>註 17</b> <math>1 - 1 = 0</math>、<math>1 - 0 = 1</math>，因此 R 的內容為 0、1 交換</p>



## 附錄三、Arcade 積木指令介紹

以下列出 Arcade 所有預設的積木指令並說明其用途，最後再介紹遊戲中常常會用到的動畫積木指令。

類別	積木指令	說明
 角色 創建 (Create)		創建一個新的角色名稱為 mySprite，並賦予一個圖像和類別（可自訂）。 預設角色類別：Player（玩家）
 角色 物理 (Physics)		設定角色持續的移動速度（velocity）。 vx：角色在水平方向的移動速度。 vy：角色在垂直方向的移動速度。
		設定角色圖像中心在屏幕中的位置，預設屏幕解析度為 160×120 像素，左上角座標為 (0,0)。
		設定角色的位置、速度、加速度、生命週期、摩擦力、邊緣、深度等屬性。（參考 3-2 節）
		增加或減少角色的位置、速度、加速度、生命週期、摩擦力、邊緣、深度等屬性。
		獲取角色的位置、速度、加速度、生命週期、摩擦力、邊緣、深度等屬性。
 角色 效果 (Effects)		設定角色的周圍產生特效，按「+」可設定特效的時間。
		解除前一積木產生的角色特效。
		刪除指定的角色。 例如子彈打中目標時，應立即將子彈刪除，按「+」可設定該角色周圍產生特效及特效持續時間。
		設定角色的上方顯示指定的文字（白底黑字），按「+」可設定該文字出現的時間。
		設定角色在屏幕中的物理行為（參考 3-2 節）

類別	積木指令	說明
<div style="background-color: #0070C0; color: white; padding: 2px;">↗ 角色</div> 發射物 (Projectiles)		由角色 mySprite 目前位置創建一個持續移動的發射物角色（參考 3-3 節），預設角色類別：Projectile（發射物）。 vx：角色在水平方向移動的速度。 vy：角色在垂直方向移動的速度。
		由屏幕側邊創建一個持續移動的發射物角色（參考 3-4 節）。預設角色類別：Projectile（發射物）。 vx：角色在水平方向移動的速度。 vy：角色在垂直方向移動的速度。
<div style="background-color: #0070C0; color: white; padding: 2px;">↗ 角色</div> 重疊 (Overlaps)		當兩個指定的角色「類別」重疊時觸發執行事件區塊內的程式，一般用於碰撞偵測（參考 3-5 節）。
		判斷兩個「角色」是否重疊，如果重疊回傳「true」，否則回傳「false」，回傳結果可提供給布林值積木或條件積木做為執行程式的依據。
		重新設定已經存在角色的類別，一般用於將發射物的類別「Projectile」改為其他類別。
<div style="background-color: #0070C0; color: white; padding: 2px;">↗ 角色</div> 生命週期 (Lifecycle)		當新創建的角色其「類別」與積木中指定的類別相同時，執行事件區塊內的程式。在事件內如欲操作新創建的角色需使用角色代稱「sprite」。
		當刪除的角色其「類別」與積木中指定的參數相同時，執行事件區塊內的程式。
<div style="background-color: #0070C0; color: white; padding: 2px;">↗ 角色</div> 圖像 (Image)		獲取指定角色的「圖像」。
		改變已經存在的角色圖像，可用於得分達到一定程度時改變角色的外觀。
		獲取指定的「類別」，可將獲取的類別存於變數中再搭配設定角色類別積木使用。

類 別	積木指令	說 明
<div style="background-color: #e67e22; color: white; padding: 5px; display: inline-block; border-radius: 10px;">  控制器                 </div> 單個玩家 (Single Player)		設定使用方向按鈕控制指定角色的移動，按「+」可設定角色的水平和垂直移動速度。
		當指定的按鈕發生動作時，執行事件區塊內的程式。 按鈕：A、B、上、下、左、右、選單、任何動作：按住、鬆開、重複
		檢查是否按下了指定的按鈕，回傳結果可提供給布林值積木或條件積木做為執行程式的依據。
		獲取按下左鍵或右鍵時的水平移動量，按「+」可將移動量乘以指定的倍率(%)。
		獲取按下上鍵或下鍵時的垂直移動量，按「+」可將移動量乘以指定的倍率(%)。
<div style="background-color: #e67e22; color: white; padding: 5px; display: inline-block; border-radius: 10px;">  控制器                 </div> 多個玩家 (Multiplayer)		設定指定的玩家可使用方向按鈕控制所屬角色移動方向，按「+」可設定角色的水平和垂直移動速度。
		設定當指定玩家（第二遊戲機）連接或斷開時，執行事件區塊內的程式。
		設定指定的玩家當指定的按鈕發生動作時，執行事件區塊內的程式。 按鈕：A、B、上、下、左、右、選單、任何動作：按住、鬆開、重複
		檢查指定的玩家是否按下了指定的按鈕，回傳結果可提供給布林值積木或條件積木做為執行程式的依據。
		獲取指定的玩家按下左鍵或右鍵時的水平移動量，按「+」可將移動量乘以指定的倍率(%)。
	獲取指定的玩家按下上鍵或下鍵時的垂直移動量，按「+」可將移動量乘以指定的倍率(%)。	

類別	積木指令	說明
● 遊戲 遊戲控制 (Gameplay)		遊戲執行中會不斷的快速觸發此事件，並重複執行此事件區塊內的程式，在此事件內如使用延遲 (Delay) 指令時，觸發事件的時間並不會變慢。
		遊戲執行時每隔指定的時間 (單位：毫秒)，會觸發此事件，並執行此事件區塊內的程式，間隔時間設定後程式執行中無法再變動。
		設定遊戲結束，並顯示輸或贏的結果文字，按「+」可設定結束時的畫面特效。
		獲取當前遊戲執行時間 (單位：毫秒)
		重置 (Reset) 遊戲，停止程式並清除所有內容，從「當啟動時」事件積木開始執行。
● 遊戲 提示 (Prompt)		在屏幕上顯示指定的提示文字，並等待使用者按「A」鍵，才會繼續程式執行，按「+」可增加一列文字。
		在屏幕上顯示指定的文字，並等待使用者按「A=OK」鍵開始於螢幕中選擇字母或數字，游標移至屏幕左下「ABC」可切換大小寫；游標移至屏幕右下「OK」並按「A」鍵才會繼續程式執行，按「+」可增加一列文字。 此積木需置於布林值積木或條件積木中做為執行程式的依據。
		獲取詢問積木回傳的數字，按「+」可設定輸入的數字位數。
		獲取詢問積木回傳的文字，按「+」可設定輸入的字元長度。
● 遊戲 對話 (Dialogs)		設定對話框的文字顏色，需置於對話內容積木之前。
		設定取代對話框內  的圖案，需置於對話內容積木之前。
		設定對話框的底圖，需置於對話內容積木之前。
		在指定的屏幕位置顯示文字對話內容，並等待使用者按「A」鍵才會繼續程式執行。















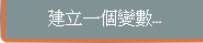
類別	積木指令	說明
 音效 音效 (Sounds)		設定播放一個短音效，並繼續執行下一積木指令。
		設定播放一個短音效，等待播放完畢才繼續執行下一積木指令。
		持續播放指定的音效。
		停止播放指定的音效。
		停止所有正在播放的音效。
 音效 旋律 (Melody)		播放自訂或素材庫中的旋律 (melody)，速度單位為每分鐘的節拍數 (bpm=beats per minute)。
 音效 音高 (Tone)		播放指定的音調 (tone) 並持續指定的時間 (節拍)。
		持續播放指定的音調 (tone)。
		休息或靜音一段指定的時間 (節拍)。
		獲取指定音符 (Note) 的頻率值。
 音效 音量 (Volume)		設定音量 (0 ~ 255)。
 音效 演奏速度 (Tempo)		加快或減慢演奏的速度 (節拍數/分鐘)。
		設定演奏的速度 (節拍數/分鐘)。
		獲取演奏的速度 (節拍數/分鐘)。
		獲取演奏的時間 (節拍)。

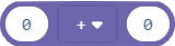
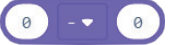
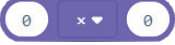
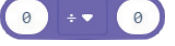











類別	積木指令	說明
 場景 畫面 (Screen)		獲取屏幕的寬度，目前預設為 160
		獲取屏幕的高度，目前預設為 120
		設定遊戲畫面底色（有透明色和 15 種顏色可選），預設為黑色。
		設定遊戲的背景圖像，可自行繪製或從素材庫中取得背景圖像，背景圖像無法移動。
		獲取遊戲畫面底色。
		獲取遊戲畫面背景圖像。
 場景 效果 (Effects)		設定遊戲開始時的畫面特效，按「+」可設定特效持續時間。
		設定遊戲結束時的畫面特效。
 場景 貼圖 (Tiles)		設定主角能自由移動的範圍，貼圖地圖大小預設為 16×16 塊貼圖，每塊貼圖固定 16×16 像素，可自訂或使用系統預設貼圖組合出遊戲地圖，地圖中可指定位置設定為牆。
		當指定的角色類別碰到牆壁時會觸發此事件，並將執行事件內的程式。
		當指定的角色類別碰到地圖中指定的貼圖時會觸發此事件，拖曳「location」可獲取當時的座標值（欄，列）。 注意：需曾在貼圖地圖中點選過的貼圖，才會出現在下拉的貼圖清單中。
		獲取地圖中所有指定貼圖的座標（欄，列），需使用陣列變數來存放獲取的座標值。 注意：需曾在貼圖地圖中點選過的貼圖，才會出現在下拉的貼圖清單中。











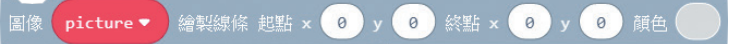





類 別	積木指令	說 明	
<p><b>場景</b></p> <p>貼圖 (Tiles)</p>		<p>將指定的角色隨機放置到指定的貼圖上，即使該位置是牆也一樣可以。 注意：需曾在貼圖地圖中點選過的貼圖，才會出現在下拉的貼圖清單中。</p>	
		<p>將指定的角色放置到指定的地圖座標（欄，列）上，即使該位置是牆也一樣可以。</p>	
		<p>當牆壁動作設為「開」時，將指定的地圖座標（欄，列）位置設定為牆。</p>	
		<p>設定指定的貼圖放置到指定的地圖座標（欄，列），可將上一事件積木中的「location」拖曳至欄列座標參數上。 注意：需曾在貼圖地圖中點選過的貼圖，才會出現在下拉的貼圖清單中。</p>	
		<p>獲取貼圖地圖中指定的座標（欄，列）值。</p>	
	<p><b>場景</b></p> <p>碰撞 (Collisions)</p>		<p>判斷「角色」的指定方向是否是指定的「貼圖」，如果是回傳「true」否則回傳「false」，回傳結果可提供給布林值積木或條件積木做為執行程式的依據。</p>
			<p>判斷「角色」是否碰到指定方向的牆，如果碰到回傳「true」否則回傳「false」，回傳結果可提供給布林值積木或條件積木做為執行程式的依據。</p>
<p><b>場景</b></p> <p>鏡頭 (Camera)</p>		<p>當使用鏡頭跟隨角色時，獲取目前遊戲畫面頂部實際是位於地圖頂部的垂直位置。</p>	
		<p>當使用鏡頭跟隨角色時，獲取目前遊戲畫面左側實際是位於地圖左側的水平位置。</p>	
		<p>將遊戲畫面固定在地圖的指定座標上。</p>	
		<p>當貼圖地圖大於屏幕範圍，遊戲主角在地圖中移動時，可能會移動超出屏幕範圍消失在鏡頭中，為避免此狀況，使用鏡頭跟隨角色，讓屏幕跟著遊戲主角移動。</p>	
		<p>畫面快速的來回移動指定像素，並持續指定的時間，讓玩家感受到碰撞效果。</p>	

類別	積木指令	說明
 資訊 得分 (Score)		獲取目前得分。
		獲取目前最高得分。
		設定得分為指定值。
		增加或減少得分值。
 資訊 生命 (Life)		獲取目前生命值。
		設定生命為指定值。
		增加或減少生命值。
		當生命值為 0 時執行事件中的程式。 如程式中無此事件，則當生命值為 0 時結束遊戲，屏幕顯示「GAME OVER」。
 資訊 倒數計時 (Countdown)		設定遊戲倒數計時。
		設定停止倒數計時。
		當倒數計時到 0 秒時，執行事件中的程式。 如程式中無此事件，則當計時到 0 秒時結束遊戲，屏幕顯示「GAME OVER」。
 資訊 多個玩家 (Multiplayer)		當指定玩家生命值為 0 時執行事件中的程式。 如程式中無此事件，則當指定玩家生命值為 0 時結束遊戲，屏幕顯示「GAME OVER」。
		判斷指定玩家是否還有生命（生命值大於 0 表示還有生命）。
		增加或減少指定玩家的生命值。
		設定指定玩家的生命值。
		獲取指定玩家的生命值。
		增加或減少指定玩家的得分值。
		設定指定玩家的得分為指定值。
		獲取指定玩家目前的得分值。

類 別	積木指令	說 明
<div data-bbox="185 922 414 974" style="background-color: #4CAF50; color: white; padding: 2px; display: inline-block; border-radius: 3px;">C 迴圈</div> (Loop)		遊戲執行中會不斷的觸發此事件，事件中的程式將持續重複執行，事件中如有暫停積木指令，將暫停到時間結束才會繼續觸發此事件。如程式中有兩個「重複無限次」積木，必須注意這兩個事件內的程式是同時在執行。
		暫停指定的時間後才會繼續執行。
		重複執行積木區塊內的程式直到指定的次數，才會離開此積木往下一個積木指令執行，此迴圈指令無法得知正在執行第幾次。
		當判斷條件（布林值積木）為 true 時，會執行積木區塊內的程式，執行完積木區塊內的程式後會再次判斷條件，只有當條件為 false 時，才會離開此積木往下一個積木指令執行。
		重複執行積木區塊內的程式，每次執行會將次數放入指定的變數中，當執行到指定的次數，才會離開此積木往下一個積木指令執行（0 到 4 會執行 5 次）。此迴圈指令可得知正在執行第幾次（存於指定的變數中）。
		使用陣列清單（list）的項目數量來決定積木區塊內程式的執行次數，會依序從陣列中取得一個項目，並將之暫存於 value 變數中，直到取完陣列最後一個項目，才會離開此積木往下一個積木指令執行。（參考 2-6 節）
		開始執行遊戲時或是重置（RESET）後，首先被觸發的事件，此積木區塊內的程式在遊戲運行中只會被執行一次，一般用來做遊戲的初始設定。

類別	積木指令	說明
 邏輯 條件 (Conditionals)		如果條件成立 (true) 則執行積木區塊內的程式，如果條件不成立 (false) 則離開此積木並往下一個積木指令執行，按「+」可增加不成立時要執行的區塊。
		如果條件成立 (true) 則執行積木上面區塊內的程式，如果條件不成立 (false) 則執行積木下面區塊內的程式，按「+」可增加下一個條件判斷式。
 邏輯 比較 (Comparison)		判斷第一數是否等於第二數，條件成立回傳 true，否則回傳 false。
		判斷第一數是否小於第二數，條件成立回傳 true，否則回傳 false。
		判斷兩字串是否相同，相同回傳 true，否則回傳 false。
 邏輯 布林值 (Boolean)		AND 邏輯運算，二個布林值均為 true 回傳 true，否則回傳 false。
		OR 邏輯運算，二個布林值只要有任一個為 true 則回傳 true，否則回傳 false。
		NOT 邏輯運算，將指定布林值反相後回傳。
		指定布林值為 true。
		指定布林值為 false。
 變數 (Variables)		變數是一個在程式中隨時可以任意給定未知或已知資訊的容器，容器中可放數字、文字、陣列、座標、角色或貼圖等資訊。





類 別	積木指令	說 明
<div style="background-color: #4a7c9c; color: white; padding: 5px; display: flex; align-items: center;"> <span style="font-size: 1.2em; margin-right: 5px;">☰</span> <span>數學</span> </div> <p style="text-align: center;">(Math)</p>		獲取兩數相加之值
		獲取兩數相減之值
		獲取兩數相乘之值
		獲取兩數相除之值
		獲取一指定數值
		獲取兩數相除之餘數
		獲取兩數之最小值
		獲取兩數之最大值
		獲取指定數之絕對值
		獲取指定數之平方根值
		獲取指定數之四捨五入值
		獲取指定數之間之隨機值
		獲取不超出最低及最高之指定數值
		獲取在第一數值範圍對應到第二範圍時指定數之對應值
		依指定的機率來隨機產生「true」，結果為「true」或「false」，回傳結果可提供給布林值積木或條件積木做為執行程式的依據。

類別	積木指令	說明
 圖像 創建 (Create)		獲取一個透明（預設為 16x16 像素）的圖像或素材庫中的圖像，如要將圖像顯示於屏幕，需創建一個角色並將圖像指定給該角色。
		獲取一個自訂大小（單位：像素）的空白（透明）圖像，如要將圖像顯示於屏幕，需創建一個角色並將圖像指定給該角色。
		獲取一個指定的圖像或圖像的複本。
		獲取畫面物件，包含整個畫面的圖像和寬、高等屬性。
		獲取一個素材庫中的圖像，如要將圖像顯示於屏幕，需創建一個角色並將圖像指定給該角色。
		將已建立的圖像填滿指定的顏色。
		獲取已建立的圖像中指定座標的顏色。
		
	<p data-bbox="448 1220 1146 1252">在已建立的圖像中，將指定的座標點設為指定的顏色。</p>	
		
	<p data-bbox="448 1371 1146 1403">在已建立的圖像中，繪製指定大小和顏色的空心矩形。</p>	
		
	<p data-bbox="448 1522 1081 1554">在已建立的圖像中，繪製指定長度和顏色的線條。</p>	
		
<p data-bbox="448 1673 1146 1705">在已建立的圖像中，填滿指定大小和顏色的實心矩形。</p>		
 圖像 轉換 (Transformations)		在已建立的圖像中尋找指定的顏色用另一顏色取代。
		將已建立的圖像垂直鏡像翻轉。
		將已建立的圖像水平鏡像翻轉。

類別	積木指令	說明
<p> 函式 (Functions)</p>	<p></p>	<p>建立函式是創建一段能夠重複利用的程式，函式的呼叫 (call) 指的是在程式的某一處透過函式的名稱來執行該段程式。 建立函式時可指定未來呼叫該函式時是否需傳參數供函式內部處理。</p>
<p> 陣列 創建 (Create)</p>	<p>  與上一積木相同。 </p>	<p>建立一個陣列變數，陣列清單中的項目值可存放文字、數值、角色或圖像，所有項目資料型態需相同，按「+」可於清單後面新增一個空白項目，按「-」可刪除清單中的最後一個項目，支援多維陣列。 產生一個陣列清單，項目數量為 0。</p>
<p> 陣列 讀取 (Read)</p>	<p>    </p>	<p>獲取指定陣列清單的項目數量。 獲取陣列清單中指定位置的項目值，清單編號 (索引值) 從 0 開始。 獲取陣列清單中指定位置的項目值，並刪除該項目值，後面項目值全部往前移動。 獲取陣列清單中最後一個項目值 (編號為陣列長度減一)，然後刪除該項目值。 獲取陣列清單中第一個項目值 (編號為 0 的項目值)，然後刪除該項目值，後面項目值全部往前移動。</p>

類別	積木指令	說明
<div style="background-color: #f4a460; padding: 5px; display: flex; align-items: center;"> <span style="font-size: 1.2em; margin-right: 5px;">☰</span> <span>陣列</span> </div> <p style="text-align: center; margin-top: 10px;">修改 (Modify)</p>		
	<p>設定陣列清單中指定編號（索引值）的項目值，清單編號從 0 開始。</p>	
		<p>在陣列清單後面新增一個指定的項目值。</p>
		<p>移除指定陣列清單中最後面的項目值。</p>
		<p>移除指定陣列清單中最前面的項目值，後面項目值全部往前移動。</p>
		<p>取得並在陣列清單中最前面插入一個項目值，原位置和之後的項目值全部往後移動。</p>
		<p>在陣列清單中最前面插入一個項目值，原位置和之後的項目值全部往後移動。</p>
		<p>在陣列清單中指定位置插入一個項目值，原位置和之後的項目全部往後移動。</p>
		<p>移除陣列清單中指定編號（索引值）的項目。</p>
		<p>獲取在陣列清單項目中與指定內容相同的項目編號（索引值）。 可使用此積木搜尋陣列清單中是否存在某項目值。</p>
		<p>將陣列清單中的所有項目值前後順序對調。</p>
		
<p>建立一個角色陣列清單，並將屏幕上符合指定角色類別的座標一一放入清單項目中。</p>		

類 別	積木指令	說 明
<p><b>T</b> 文字 (Text)</p>		獲取一個指定的文字，預設為空字串。
		獲取指定文字的字元長度。
		獲取所有指定文字所組合的新字串。
		獲取指定字串前面的數字部分，如指定字串的第一個字元就不是數字，則回傳「NaN」（表示未定義）。
		獲取一份陣列清單，其項目值是將指定字串（this）依分隔符號分割後的結果。
		判斷指定字串（this）內是否包含後面的文字，如果有回傳「true」否則回傳「false」。
		獲取指定字串（this）中子字串的位置（第一個字元位置編號為 0），如指定字串中無相同的子字串，則傳回 -1。
		判斷指定字串（this）是否為空值（NULL），如果是空值回傳「true」否則回傳「false」。
		獲取指定字串（this）內的部分字串內容，可以設定要開始擷取的字元位置與總共要擷取的字元數量，第一個字元編號為 0。
		獲取兩字串比較的結果（以 ASCII 做比較），結果如下： 1：前面字串比較大 0：兩字串相同 -1：後面字串比較大
		獲取指定字串（this）中指定編號（索引值）的字元，第一個字元編號為 0。
		將數字轉為文字類型，例如「角色說出」積木中的參數必須是文字類型，如果要讓角色說出數字，必須用此積木將數字轉成文字類型。
		獲取指定 ASCII 碼對應的字元。

類別	積木指令	說明
 控制台 (Console)		傳出文字資料給系統監控器，可使用系統監控模擬器以文字查看程式中即時傳出的文字資料。
		傳出文字和數值資料給系統監控器，可使用系統監控模擬器以曲線圖顯示程式中即時傳出的文字和數值資料。
 動畫 動畫 (Animate)		※ 使用「動畫」積木需擴展「animation」 (參考第 8 章)  使用多張圖像創建一個新的動畫，動畫間隔時間單位毫秒，可設定循環播放動畫。使用時需創建一個角色並將動畫名稱指定給角色的圖像屬性。
		設定動畫圖像在屏幕中的移動方式
		設定指定的角色動畫停止播放
		獲取數張影像串列
 動畫 (Legacy)		※ Legacy 動畫在同一角色中，可依不同的情境播放不同的子動畫 (參考第 8 章) 可設定子動畫情境，預設屬性有： Walking (走路)、Idle (停止)、Jumping (跳躍) 及自訂情境，動畫間隔時間可獨立設定，使用前需先創建一個角色。
		加入一個影格到指定的情境子動畫中。
		將情境子動畫附加到角色中。
		設定播放指定的情境子動畫。
		獲取目前的動畫情境。

# WiFiBoy Arcade Pro 編程學習遊戲機 (D51)



產品編號：3024012  
建議售價：\$1,480

WiFiBoy Arcade Pro 編程學習遊戲機 (D51) 是台灣在地設計、製造，講究設計質感與安全設計。除了支援微軟的 MakeCode Arcade 開發環境，還提供 WiFiBoy Python Playground 開發環境，也提供 Arduino IDE for WiFiBoy Arcade 版本，可再延伸學習電子設計與創課專案。

## 更好玩的 Scratch；更簡單的 micro:bit



### Maker 指定教材

深入淺出 用 MakeCode Arcade  
玩程式設計第一次就上手—  
使用 WiFiBoy Arcade 編程學習遊戲機  
書號：PN030  
作者：康文耀  
建議售價：\$350

### 產品規格

搭配編程軟體	MakeCode Arcade、MicroPython、Arduino	
處理器	晶片	SAMD51
	主頻	120Mhz
記憶體	Flash	512KB Flash + 4/8 MB SPI-Flash
	RAM	192KB
擴充口	12pin 2.54mm 排母座	
按鍵	6mm 彩色訂製遊戲級按鍵	
外殼	三明治膠合設計	
尺寸	10cm x 4cm x 1cm	
重量	32.5g	
配件	MicroUSB 線 x 1	

### 選配



環形彩色 LED 燈板  
產品編號：3024013  
建議售價：NT\$200

※ 價格 · 規格僅供參考 依實際報價為準

書名

# 深入淺出用MakeCode Arcade 玩程式設計第一次就上手 使用WiFiboy Arcade 編程學習遊戲機

書號  
版次

PN030  
109年9月初版

編著者  
校閱者  
總編輯  
責任編輯  
校對次數  
版面構成  
封面設計

康文耀  
楊岱霖  
張忠成  
淼清文文教 游淇文  
8次  
楊蕙慈  
楊蕙慈

出版者  
門市地址  
電話  
傳真  
網址  
電子郵件

台科大圖書股份有限公司  
24257新北市新莊區中正路649-8號8樓  
02-2908-0313  
02-2908-0112  
tkdbooks.com  
service@jyc.net

版權宣告

## 有著作權 侵害必究

本書受著作權法保護。未經本公司事前書面授權，不得以任何方式（包括儲存於資料庫或任何存取系統內）作全部或局部之翻印、仿製或轉載。

書內圖片、資料的來源已盡查明之責，若有疏漏致著作權遭侵犯，我們在此致歉，並請有關人士致函本公司，我們將作出適當的修訂和安排。

郵購帳號  
戶名

19133960  
台科大圖書股份有限公司  
※郵撥訂購未滿1500元者，請付郵資，本島地區100元 / 外島地區200元

客服專線

0800-000-599

網路購書



PChome商店街  
JY國際學院



博客來網路書店  
台科大圖書專區

各服務中心

總公司 02-2908-5945 台中服務中心 04-2263-5882  
台北服務中心 02-2908-5945 高雄服務中心 07-555-7947



線上讀者回函  
歡迎給予鼓勵及建議  
tkdbooks.com/PN030

國家圖書館出版品預行編目資料

深入淺出用MakeCode Arcade玩程式設計  
第一次就上手:使用WiFiboy Arcade編程學習  
遊戲機 / 康文耀編著.

-- 初版. -- 新北市 : 台科大圖書, 民109.09  
面; 公分

ISBN 978-986-523-114-9(平裝)

523.38 109012993

校閱 楊岱霖 · 慧燈中學教師