

高小增潤 編程教育課程單元



教育局
課程支援分部
科技教育組
二零二三年六月

小四
學生版本
第二冊

引言

教育局積極推動普及創科教育，透過持續在中小學課程加入創科學習元素，從小加強培養學生學習資訊科技和創新科技的興趣和能力，裝備學生二十一世紀所需的能力，啟發創意潛能。

配合加強創新科技教育，教育局推出「高小增潤編程教育課程單元」，供學校採用。課程單元亦配合 2020 年公布更新的《計算思維—編程教育：小學課程補充文件》而設計，協助教師把創新科技元素更有組織地融入課堂學習。學校需按「高小增潤編程教育課程單元」的內容，適當規劃課程，為所有高小學生每年推行 10 至 14 小時的增潤編程教育，以進一步培養學生的計算思維和加強創科學習。

「高小增潤編程教育課程單元」改編自香港賽馬會慈善信託基金策劃和捐助的「賽馬會運算思維教育」計劃的教材，而該計劃亦由香港教育大學、美國麻省理工學院及香港城市大學聯合策劃。教育局感謝香港賽馬會慈善信託基金與本局協作，整理及總結學校從這計劃積累的經驗，以發展「高小增潤編程教育課程單元」，供全港公帑資助學校採用。教育局課程支援分部科技教育組聯同香港教育大學數學與資訊科技學系，根據上述計劃的成果和經驗開發本課程單元。課程單元的內容已向課程發展議會科技教育委員會蒐集意見，並獲得委員會支持。

「高小增潤編程教育課程單元」涵蓋編程和計算思維的基本概念包括抽象化、算法和自動化，以及連接實物、運用感測器和執行器與環境進行互動等，讓學生透過學習編程以培養他們計算思維，以及學習創科的興趣和能力。

本小四課程單元是「高小增潤編程教育課程單元」的首個課程單元（小五和小六將陸續推出），重點建立學生編程和計算思維的基本概念穩固的基礎；透過編程學習活動建立有系統的思考和解難能力，培養計算思維。課程單元共有 8 個單元，當中核心單元 6 個，另設 2 個延伸單元供學校選用，讓能力較高或對編程具濃厚興趣的學生有增潤學習的機會，加深他們對編程及創新科技的認識。課程單元亦設專題習作，讓學生發揮計算思維能力和創意，善用編程和創新科技到不同的情境，以設計解決日常生活問題的方案，造福人群。

每級課程單元的建議課時（除延伸單元部分）為 14 小時。有關小四課程單元、建議課時安排和教學法，可參考表 1 和附錄。

表 1 小四課程單元和建議課時安排

單元	單元名稱	核心單元		延伸單元	
		建議課時 (分鐘)	課節 (每節 35 分鐘)	建議課時 (分鐘)	課節 (每節 35 分鐘)
1	簡介 Scratch 編程環境	70	2		
2	海底探險	70	2		
3	說故事	70	2		
4	太空漫遊	105	3		
5	創作迷宮遊戲	140	4		
6	運用 micro:bit 創作迷宮遊戲			70	2
7	運用 Scratch 畫圖形	105	3		
8	線條圖案藝術設計			70	2
	專題習作	280	8		
		840 (14 小時)	24	140	4

歡迎學校對「高小增潤編程教育課程單元」提出意見和建議，來函請寄：

香港九龍塘沙福道19號

教育局九龍塘教育服務中心西座1樓101室

教育局課程支援分部科技教育組

總課程發展主任（科技教育）收

傳真：2768 8664

電郵：teched@edb.gov.hk

教學法

基於科技教學內容知識 (TPACK) 的理論架構，教師可參照 TPACK 架構的七步曲教授計算思維。科技內容知識 (TCK) 是指關於使用方塊的編程環境進行編程的知識。內容知識 (CK) 是關於要教授的計算思維概念、實踐和態度的知識。教學內容知識 (PCK) 是指不涉及使用編程環境來教授內容知識 (CK) 的教學法。科技教學內容知識 (TPACK) 則是整合編程環境和教學法於一個情境來教授內容知識 (CK)。

運用上述四個維度，教師可運用七步曲的教學法來教授每一個單元，藉此培育學生的解難能力及數碼創意。最後三個步驟強調運用編程環境的科技知識啟發學生的數碼創意，再回顧審視鞏固所習得的計算思維概念、實踐和態度的內容知識(CK)，以及反思教學法(PCK) 並藉此精進教授內容知識(CK)的教學實踐 (Kong, Lai & Sun,2020; Kong & Lai, 2022; Kong, Lai & Li, 2023)。

第 1 步： TCK (介紹該單元運用編程環境的元件)

第 2 步： CK (介紹該單元需要教授的計算思維概念、實踐和態度的內容知識)

第 3 步： PCK (透過教學法如在開始編碼前先試玩該單元的遊戲或應用程式，並藉此思考該單元的遊戲或應用程式的設計。進行不插電活動，幫助學生了解編碼中較難理解的概念、實踐和態度)

第 4 步： TPACK (教師在特定情境中運用編程環境的某些元件及合適的教學法發展學生計算思維的內容知識 (CK))

第 5 步： TCK (鼓動學生提出該單元運用編程環境的元件應用到其他可能應用的情境，並藉此激發學生的數碼創意)

第 6 步： CK (協助學生反思所習得的計算思維概念、實踐和態度，鞏固所學)

第 7 步： PCK (教師自我反思單元教學中所採用的教學法，檢視如何改進下一次的教學)

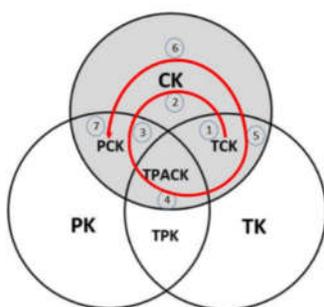


圖 1：陰影區域 (CK、TCK、PCK 和 TPACK) 的七步曲是教師在教授計算思維內容知識時的步驟 (Kong, Lai & Sun, 2020)。

參考資料

教育局：《計算思維－編程教育：小學課程補充文件》修訂版 2020 年

Kong, S. -C., & Lai, M. (2022). A proposed computational thinking teacher development framework for K-12 guided by the TPACK model. *Journal of Computers in Education*, 9(3), 379-402.

Kong, S. -C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872.

Kong, S.-C., Lai, M., & Li, Y.G. (2023). Scaling up a teacher development programme for sustainable computational thinking education: TPACK surveys, concept tests and primary school visits. *Computers & Education*, 194, 104707.

高小增潤編程教育課程單元(小四)

編輯委員會成員

江紹祥教授

香港教育大學
電子學習與數碼能力研究講座教授
數學與資訊科技學系
人工智能及數碼能力教育中心總監
教學科技中心總監

蕭家鈺女士

香港教育大學
教學科技中心
教育發展經理

馬韵斯女士

香港教育大學
教學科技中心
助理教育發展經理

丁韻怡小姐

香港教育大學
教學科技中心
助理教育發展經理

盧海林先生

香港教育大學
教學科技中心
教育發展主任

第二冊目錄

單元	單元名稱
5	創作迷宮遊戲
6	運用 micro:bit 創作迷宮遊戲 (延伸單元)
7	運用 Scratch 畫圖形
8	線條圖案藝術設計 (延伸單元)
	專題習作

單元五：創作迷宮遊戲

學習指引

目錄

第一教節

玩一玩	S5-1
想一想	S5-3
想一想、來編程	
熊貓在起點	S5-5
用鍵盤控制熊貓	S5-6
使熊貓行走	S5-7

第二教節

想一想	
分支 / 選擇 (如果熊貓碰到...)	S5-10
來編程	
碰到迷宮	S5-12
想一想、來編程	
循環	S5-14
碰到目標	S5-15

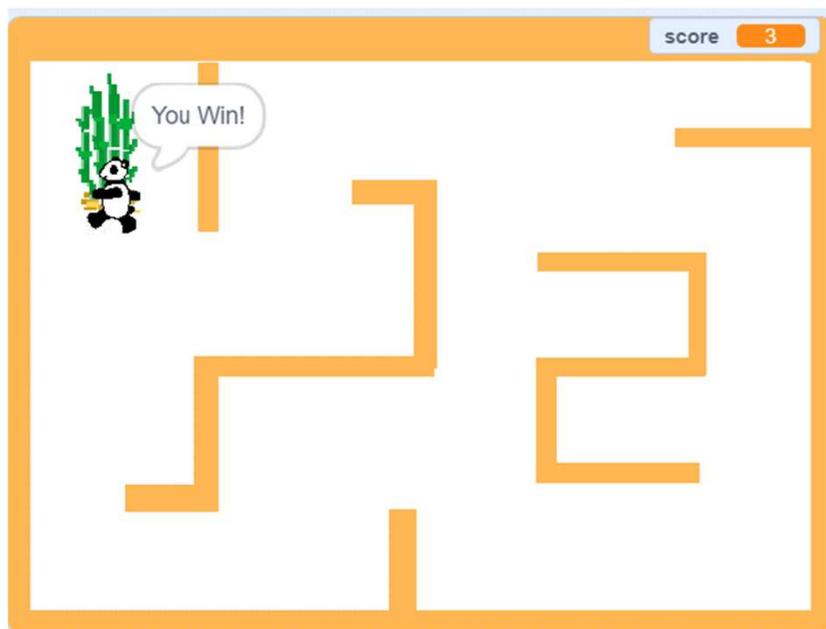
第三教節

想一想、來編程	
加入分數	S5-17
來編程	

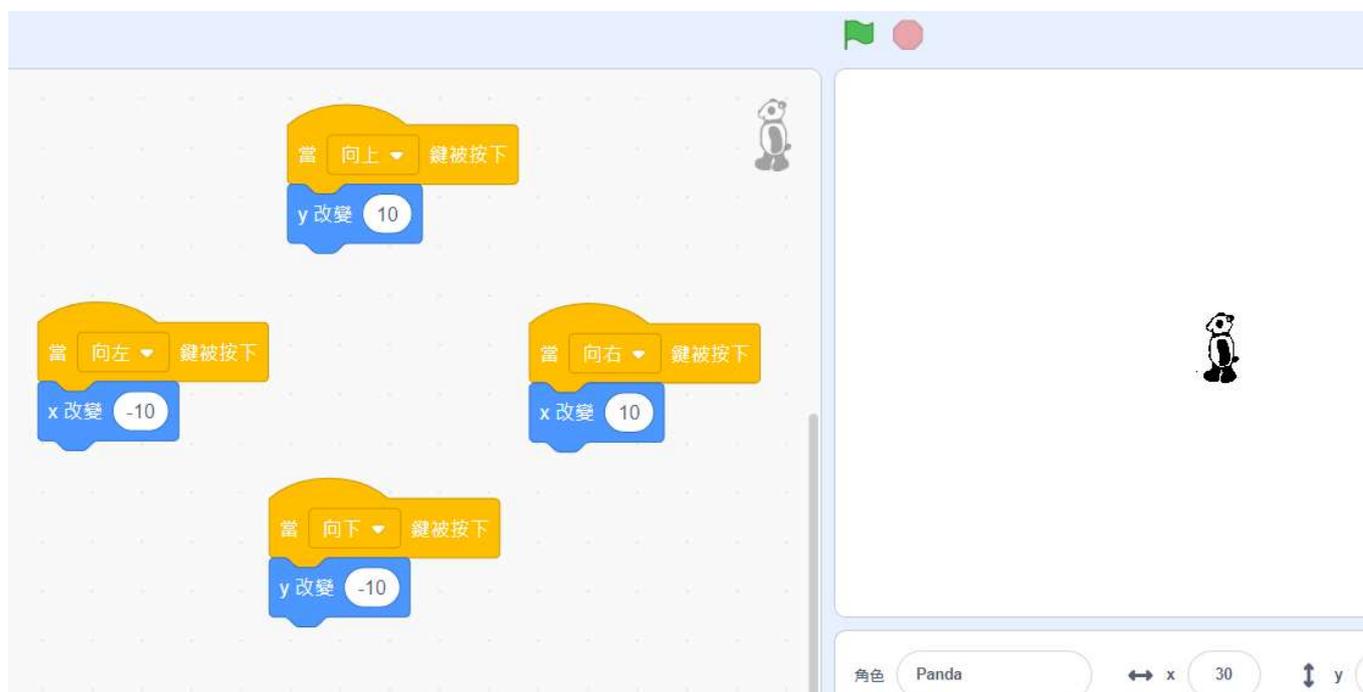
加入分數	S5-20
第四教節	
同創作	S5-22
齊反思	S5-24
重溫練習	S5-25
重溫主要元件	S5-27
重溫主要概念和實踐	S5-28
附錄 - 操作指南	S5-32

創作迷宮遊戲

讓我們學習用
Scratch 製作遊戲！



玩一玩



創作迷宮遊戲

玩一玩

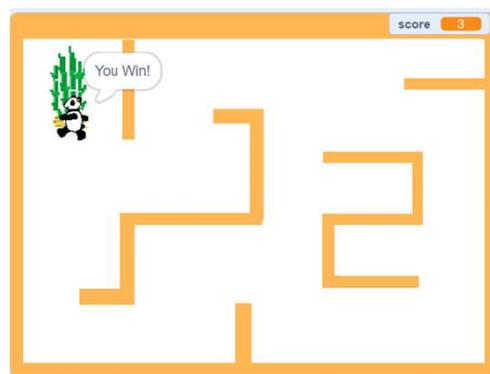
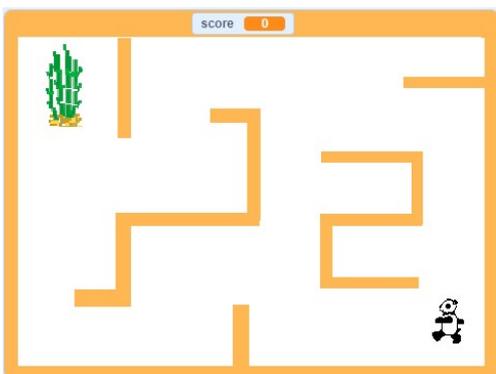
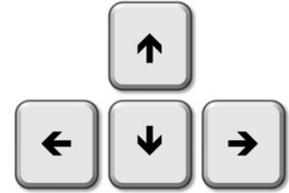
使用**鍵盤** (上 / 下 / 左 / 右鍵) **把玩**迷宮遊戲：

<https://scratch.mit.edu/projects/722154863>

如何能**勝出**遊戲呢？

當你**碰到牆壁**時發生了什麼？

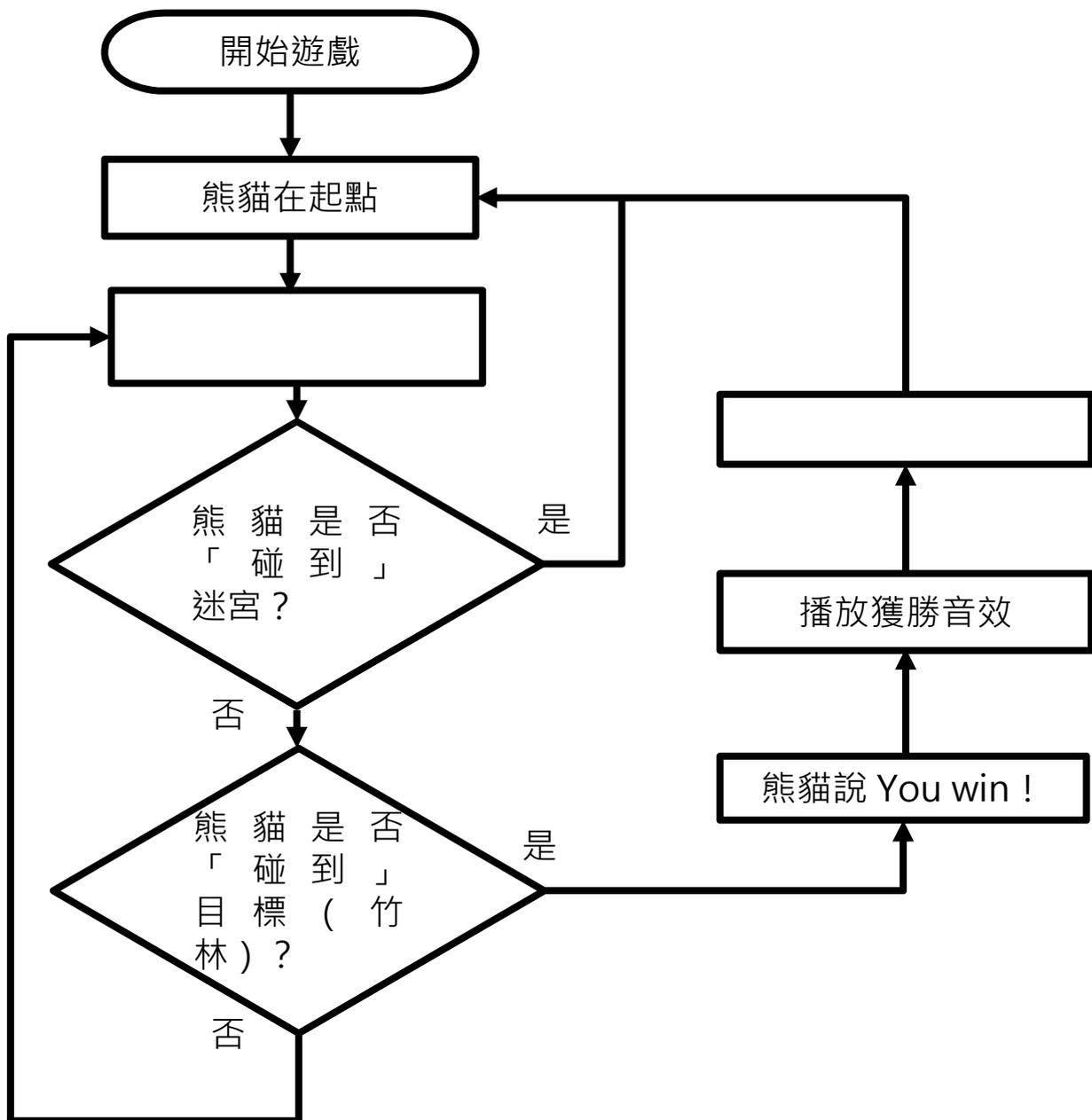
當你碰到**竹林**時發生了什麼？



創作迷宮遊戲

想一想

完成以下流程圖。



創作迷宮遊戲

從這裏開始

1. 請在 <https://scratch.mit.edu/> 登錄你的帳戶。

1

2. 打開迷宮遊戲範例：<https://scratch.mit.edu/projects/727439171> 並點擊改編按鈕。

2



3. 將專案重新命名為 Maze Game。

3

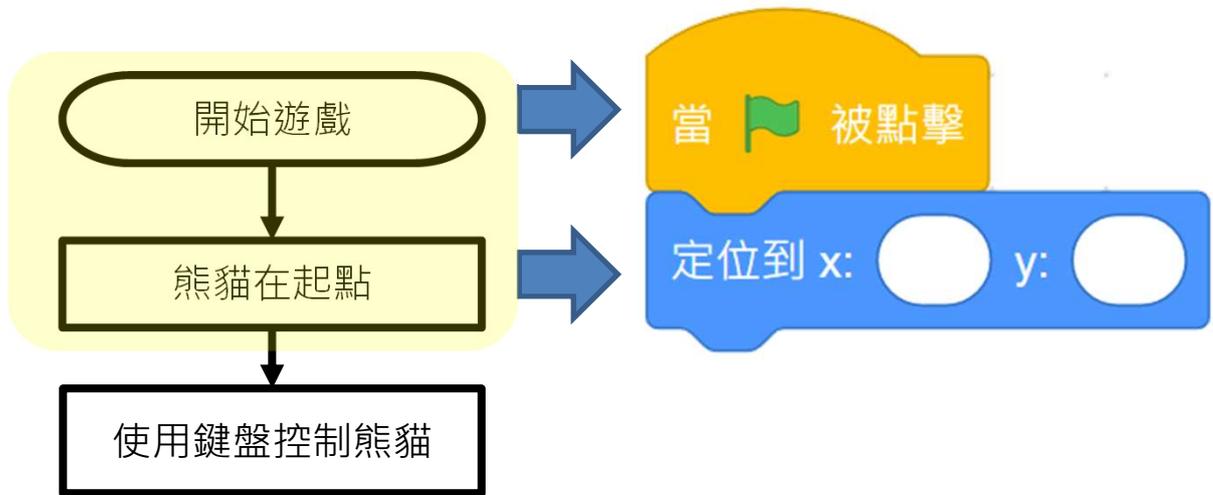
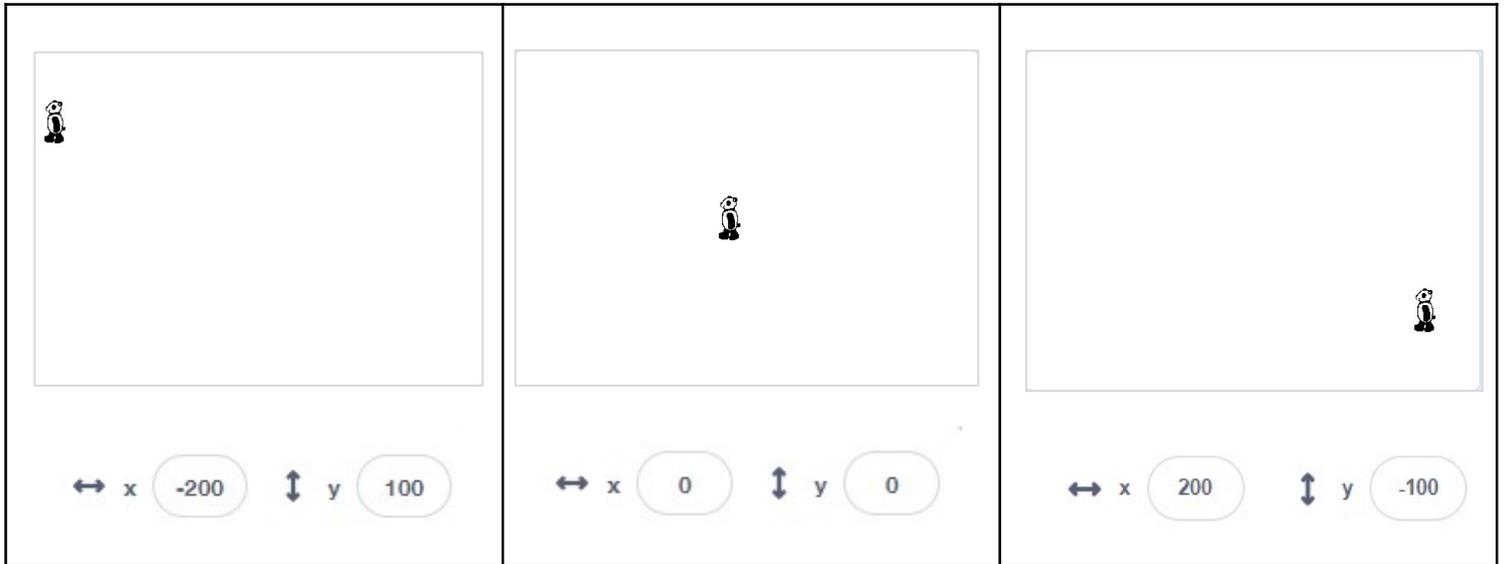


創作迷宮遊戲

想一想、來編程：熊貓在起點

可參考附錄
P.33

三隻熊貓有何分別？



🐛 測試及除錯

點擊綠旗，看看熊貓是否在起點。



創作迷宮遊戲

單元五
學習指引：第一教節

想一想、來編程：用鍵盤控制熊貓

可參考附錄
P.33

留意指令方塊和背景：

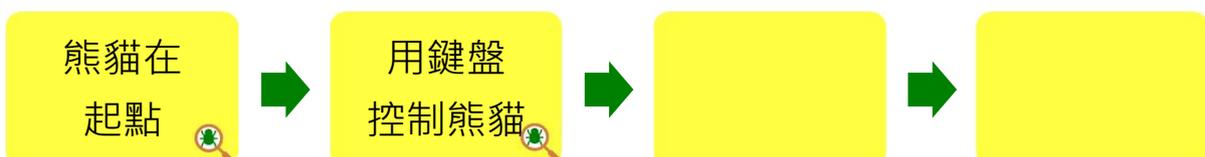


現在修改指令方塊以控制熊貓移動。



測試及除錯

你能用四個方向鍵來控制熊貓嗎？

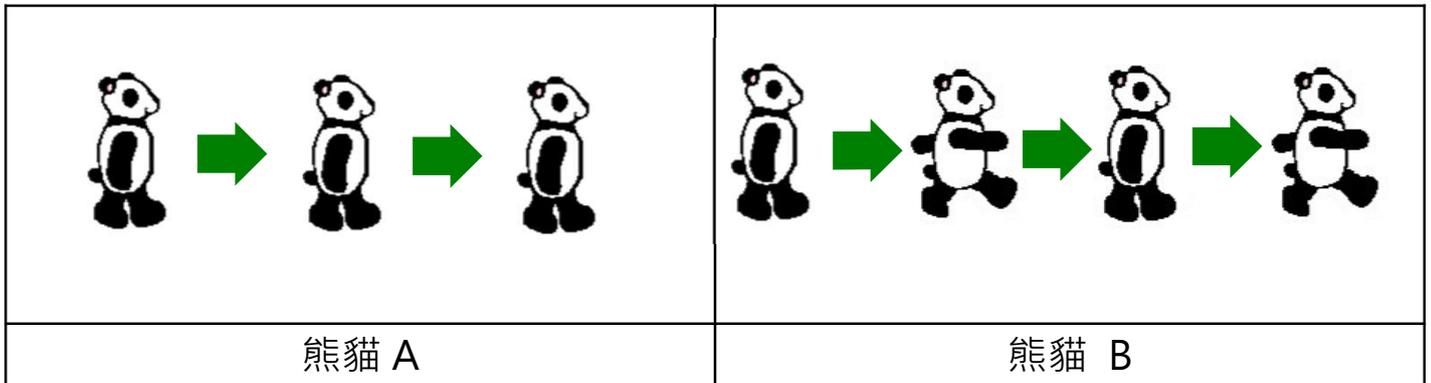


創作迷宮遊戲

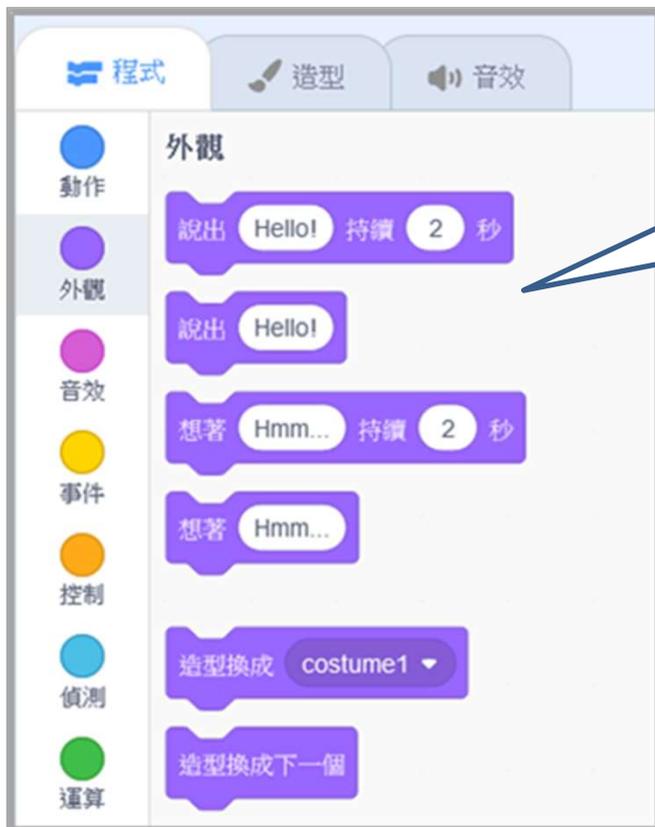
想一想、來編程：使熊貓行走

可參考附錄
P.34

哪隻熊貓看似正在「行走」？



你將如何更改角色的造型使它看起來更像在行走？

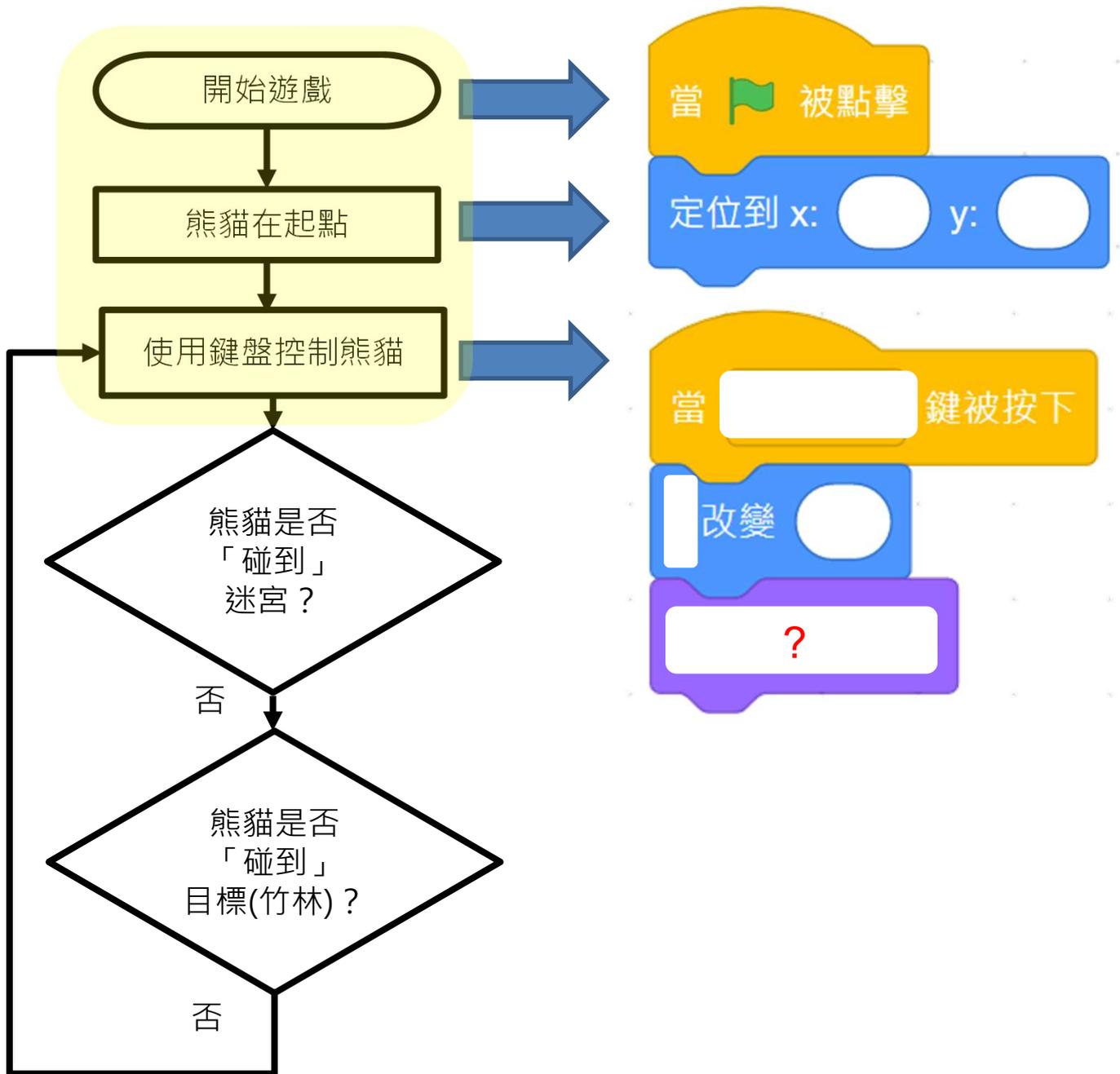


哪個指令方塊用來
使熊貓行走？



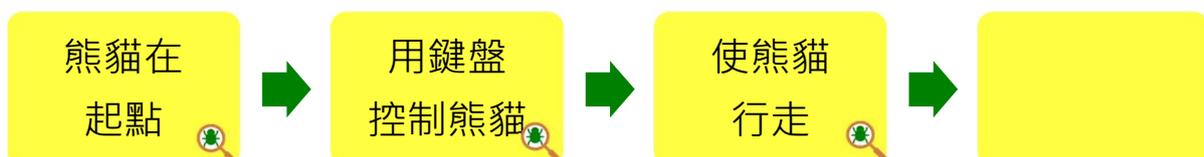
創作迷宮遊戲

想一想、來編程：使熊貓行走



測試及除錯

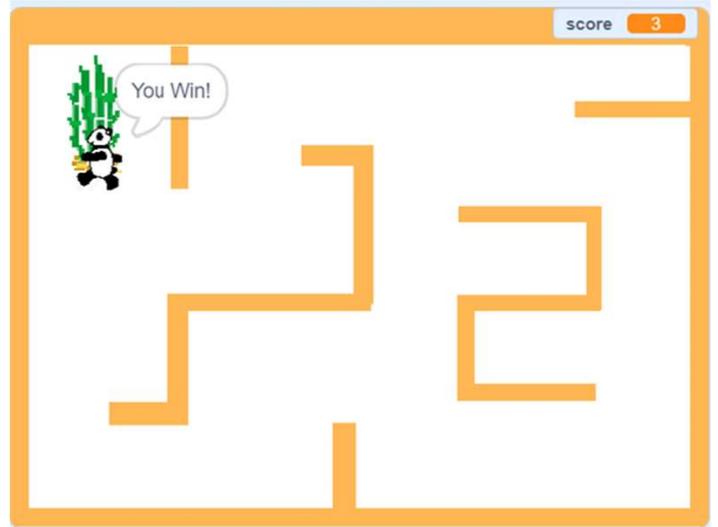
測試一下，看看熊貓能否行走。



創作迷宮遊戲

一起完成我們的遊戲吧！

你在這個遊戲會使用方向鍵引導熊貓到達它的竹林，避開牆壁。到達竹林獲得 + 1 分，撞牆則回到起點。此外，你還可以添加語音、得分和聲效。



從這裏開始

1. 請在 scratch.mit.edu 上登錄你的帳戶。

1

2. 前往 [我的東西](#) 並開你的迷宮遊戲專案。

2

3. 點擊網站右上角的 [切換到程式頁面](#) 按鈕，並繼續開發你的遊戲。

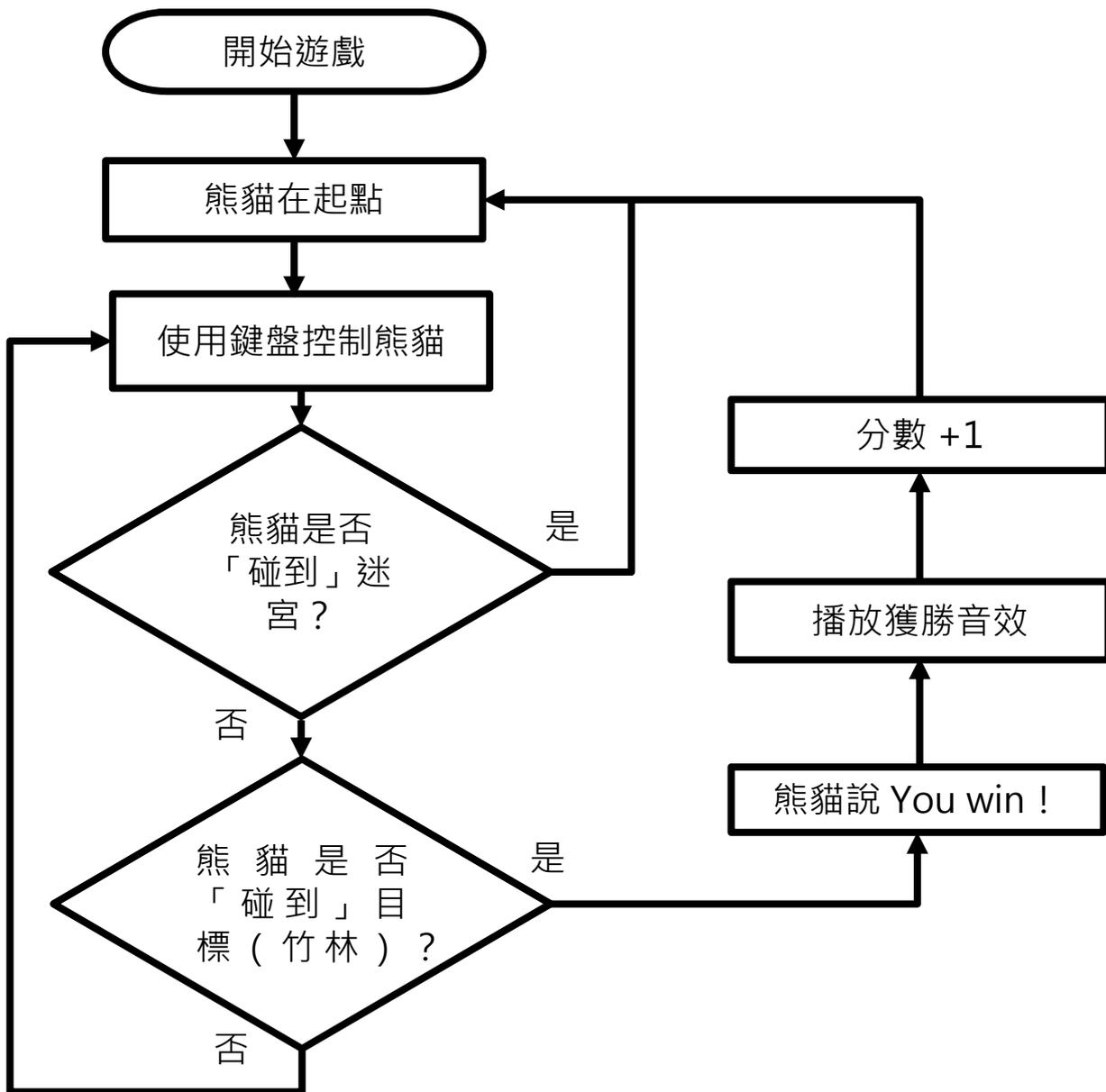
3



創作迷宮遊戲

想一想：分支 / 選擇（如果熊貓碰到...）

熊貓在遊戲中會遇到什麼事情？



創作迷宮遊戲

想一想：分支 / 選擇 (如果熊貓碰到...)

我們在範例裏有三個角色：



它們會分別做什麼事？

熊貓在走路...

如果 碰到 ?

那麼 (會發生什麼?)

如果 碰到 ?

那麼

創作迷宮遊戲

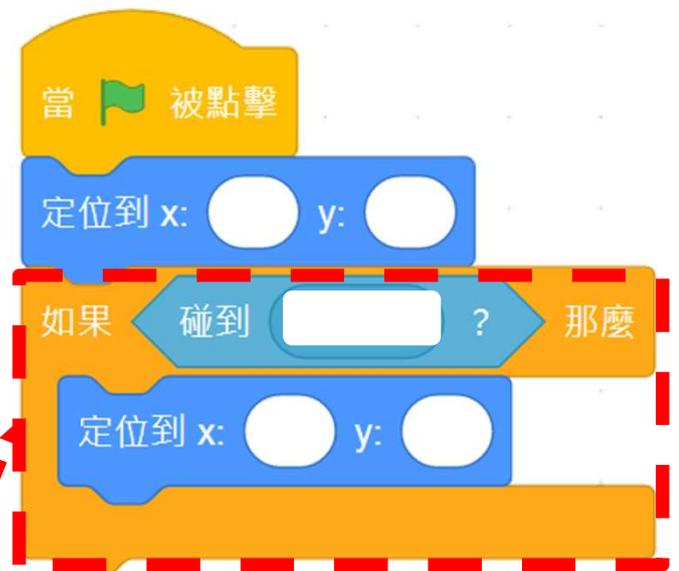
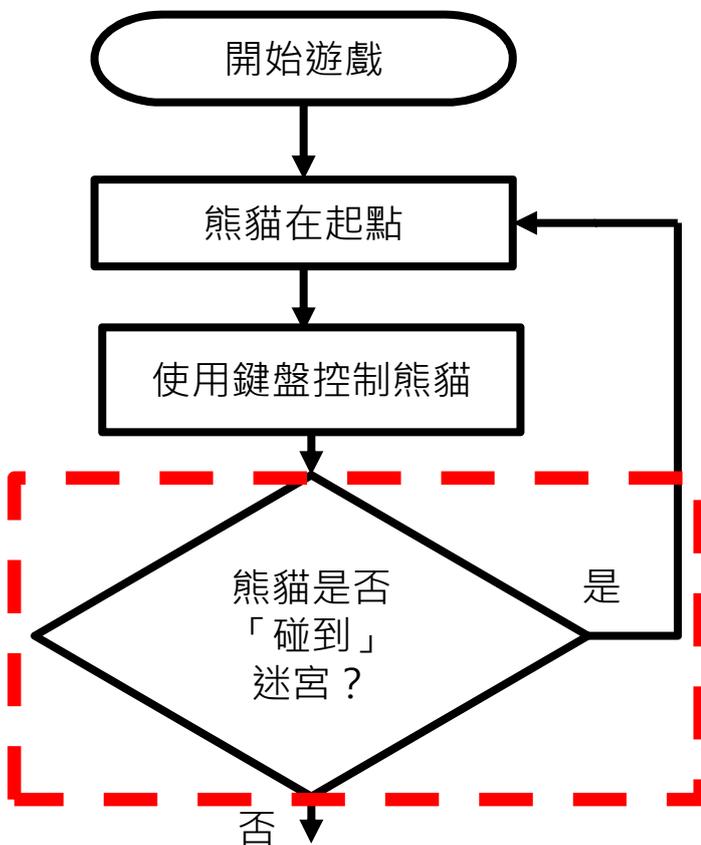
來編程：碰到迷宮

可參考附錄
P.35

1. 我們在整個遊戲過程希望熊貓角色在碰到某物時作出不同的行為。為此，我們應該從「控制」欄中拉出「如果-那麼」指令方塊。
2. 運用你設計的算法，哪些條件會觸發這些動作？
提示：看一下「偵測」欄。



3. 如果熊貓碰到迷宮：



重溫分支 / 選擇

1. 我們在編程使用條件句式進行推理，讓電腦做_____。
2. 條件句式總有「_____」的部分，它告訴程式當條件成立時，「_____」應該做什麼。



測試及除錯

是時候測試了！點擊綠旗並移動熊貓撞牆壁。它會回到起點嗎？
再做一次，現在有什麼不同？

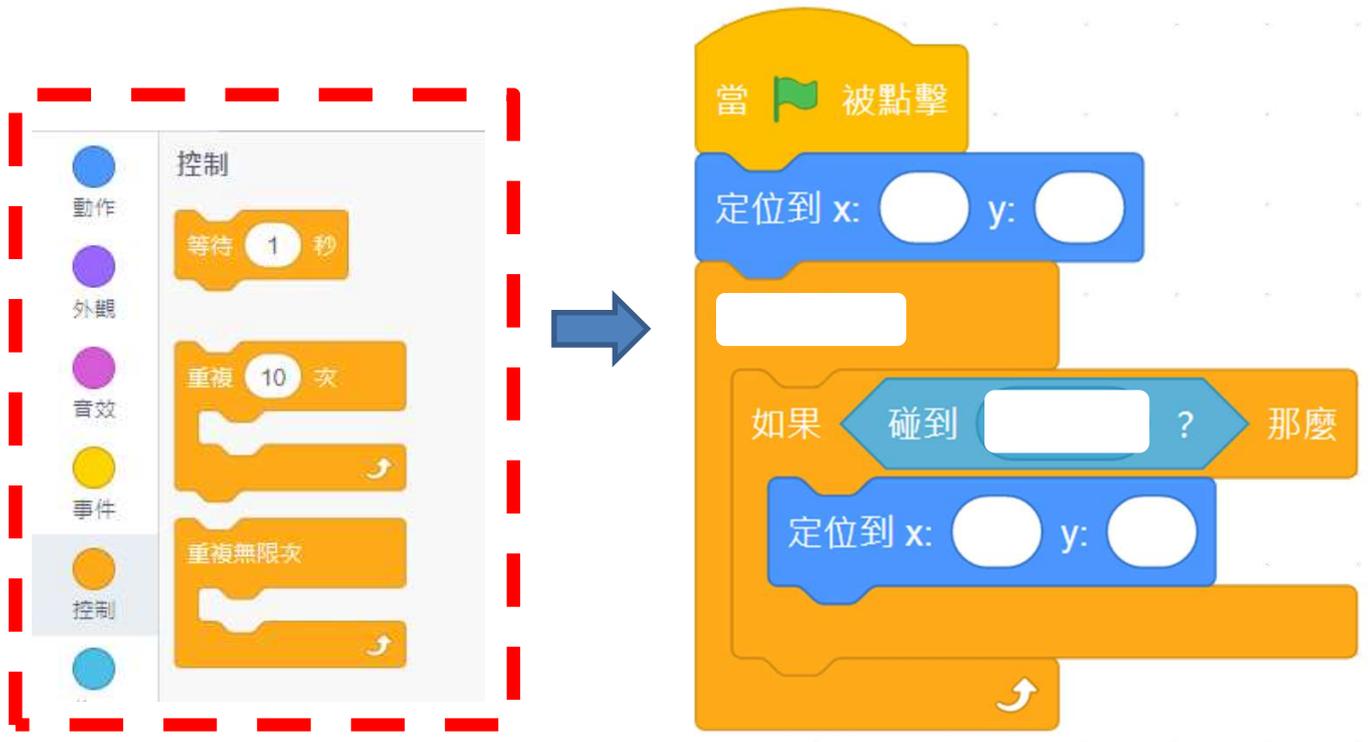


創作迷宮遊戲

想一想、來編程：循環

參考「控制」欄，你會用哪些指令方塊來解決問題呢？

可參考附錄
P.36



每次熊貓走路時，你需要不斷檢查嗎？

需要

不需要

原因：_____

測試及除錯



創作迷宮遊戲

來編程：碰到目標

如果熊貓碰到目標（竹林）：

可參考附錄
P.37

1. 讓熊貓角色說點什麼。

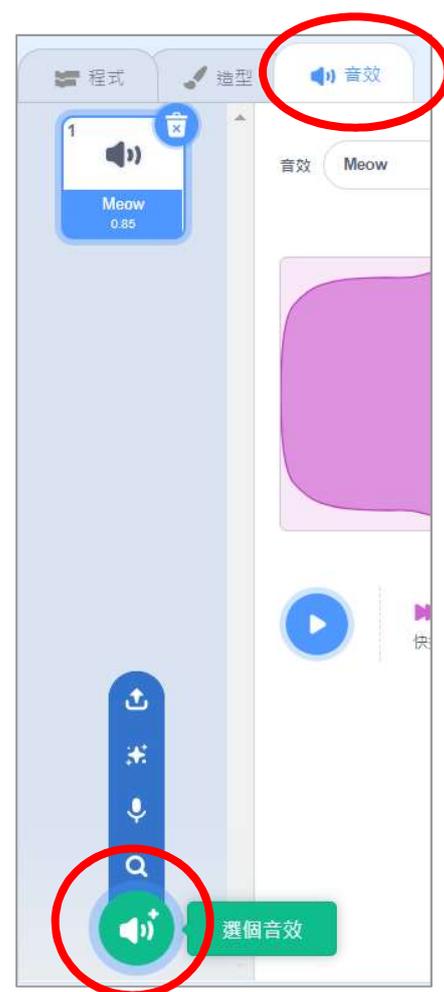
查看「外觀」欄。你會用哪一個方塊？



2. 加入勝利音效。請轉到「音效」標籤。點擊左下角的「選個音效」圖標以選擇聲音。

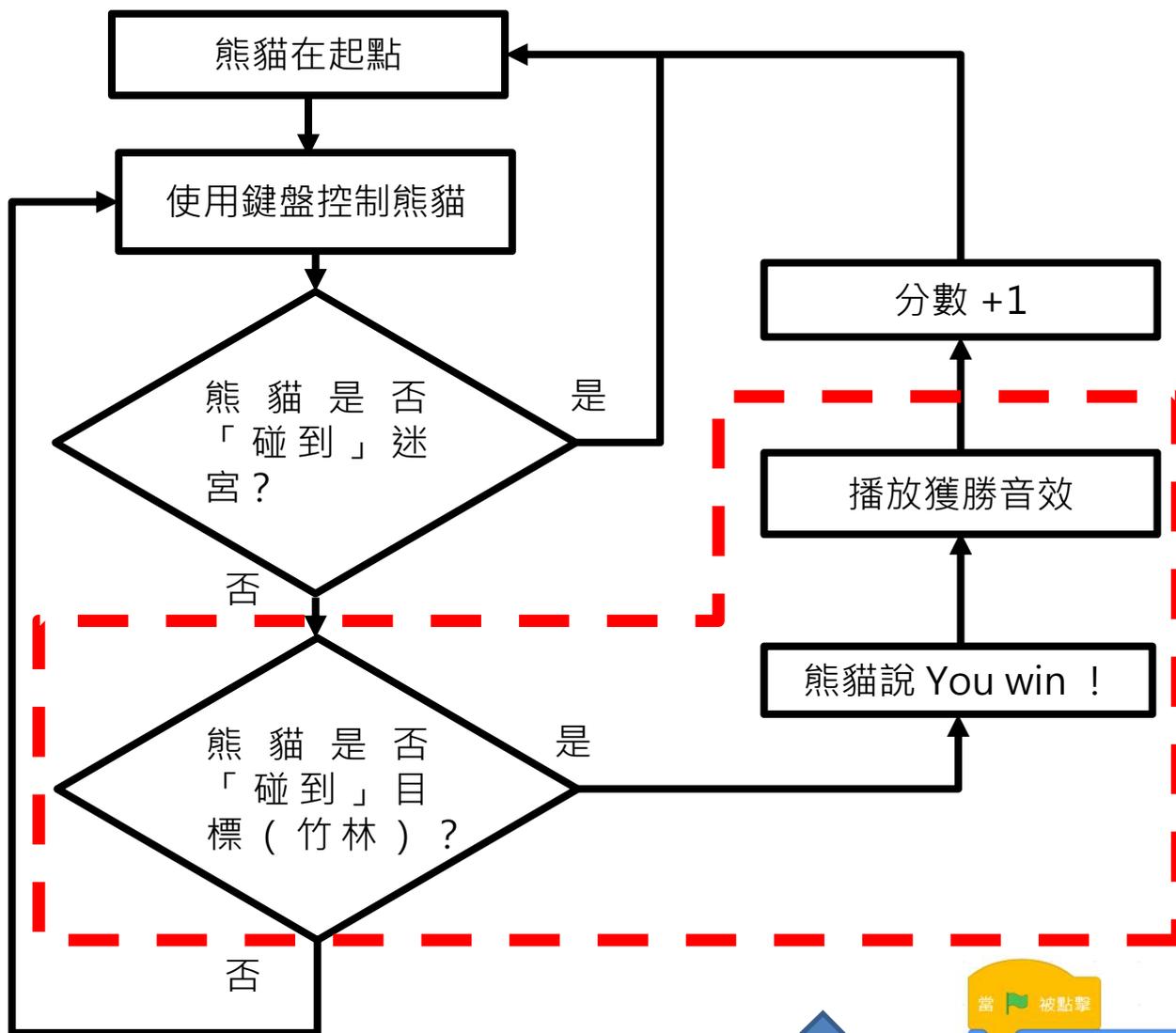
3. 你還記得如何讓你的角色播放你找到的聲音嗎？

提示：到「程式」標籤下的「音效」欄。



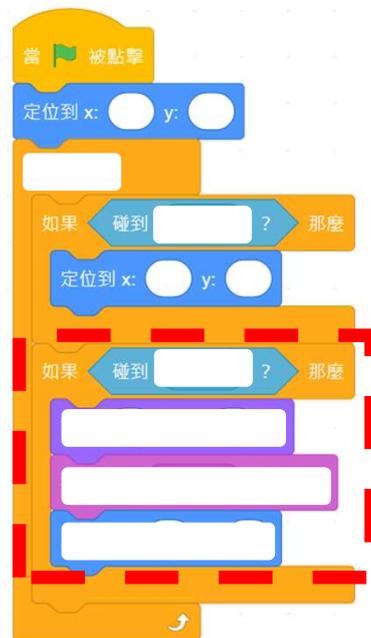
來編程：碰到目標

如果熊貓碰到目標（竹林）：



測試及除錯

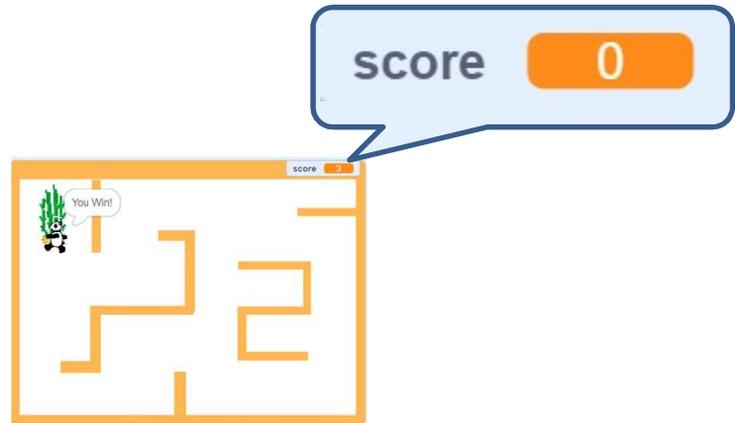
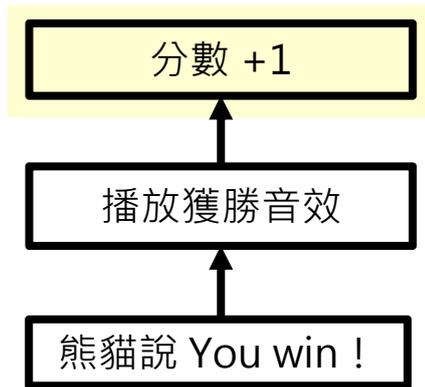
是時候測試了！盡量把熊貓移動到竹林。
緊記不斷檢查條件。



創作迷宮遊戲

想一想、來編程：加入分數

你還記得如果熊貓找到竹子我們會得到什麼嗎？是的！現在是時候在你獲勝時添加分數 (score) 了。



1. 添加一個變數用作記錄分數，轉到「變數」欄並點擊「建立一個變數」。

1



2. 將新的變數命名為 score (分數) 。

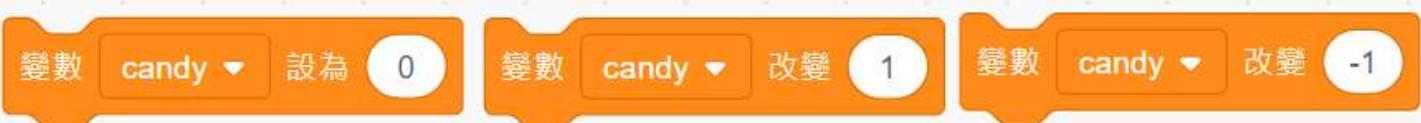
2



創作迷宮遊戲

不插電活動：變量（變數）

	<p>教師拿出一個盒子， 在盒子上寫上「糖果」。</p>
---	----------------------------------



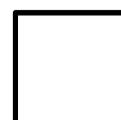
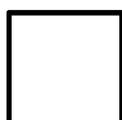
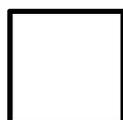
變數 candy ▾ 設為 0 變數 candy ▾ 改變 1 變數 candy ▾ 改變 -1

遊戲 1：你可以選擇其中一個，然後教師會做動作。

1. Set candy to "5"（糖果設為 5）
2. Change candy by "-2"（糖果改變 -2）
3. Change candy by "2"（糖果改變 2）

遊戲 2：猜猜盒子裏有多少顆糖果：

		
---	---	---



創作迷宮遊戲

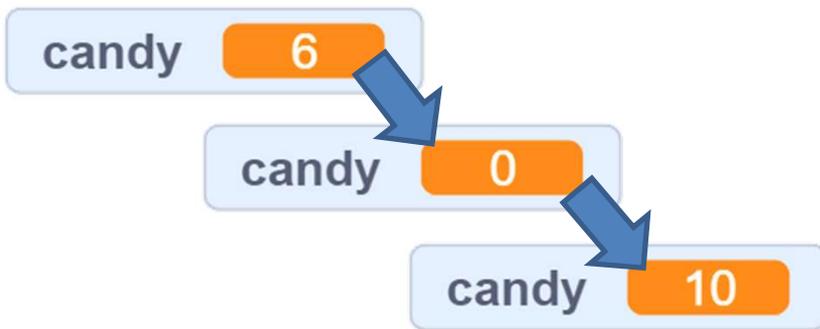
知識建立：變數

變數用於儲存數值，它具有以下屬性。試填寫空格：

1. 變數有_____。



2. 一個變數每次只能存儲_____個值。



3. 變數數值可被_____。

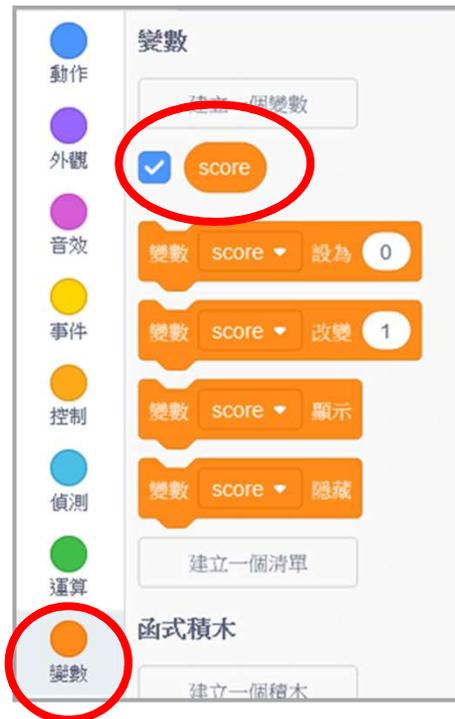


創作迷宮遊戲

來編程：加入分數

可參考附錄
P.38

1. 首先，我們需要顯示分數的變數（勾選 score 以顯示的分數），並在遊戲開始時將分數重置為 0。
提示：查看「變數」欄以決定應該添加哪個方塊來重置分數。



2. 當熊貓到達竹林時，分數該如何變化？

測試及除錯

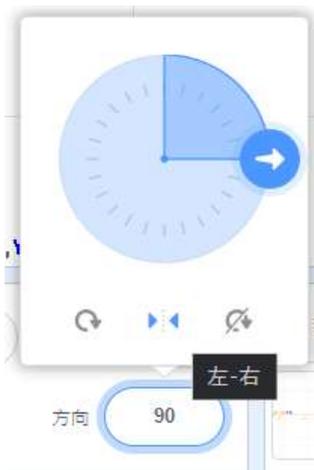
你做到了！再次玩迷宮遊戲，看看一切是否正常。



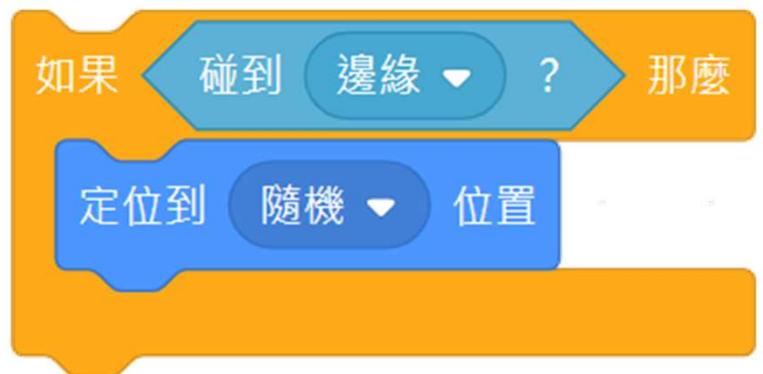
創作迷宮遊戲

程式指令參考：

退後時改變熊貓臉的方向：



添加邊界/ 回彈障礙物等：

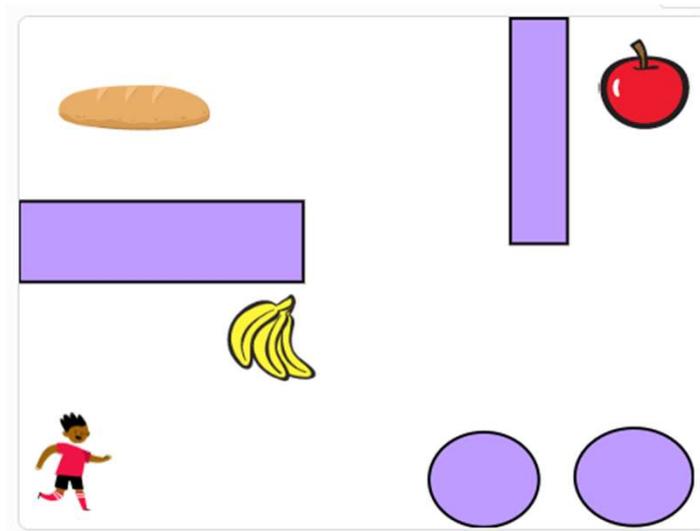


創作迷宮遊戲

同創作：超市迷宮

1. 你可發揮創意，自由創作另一個新的迷宮遊戲，或參考超市迷宮範例。
2. 以下是另一個類似迷宮遊戲的專案：控制男孩在超級市場中找麵包，如果男孩正在「碰到」麵包，就說“ I find it ! ”（我找到了！）。

參考：<https://scratch.mit.edu/projects/731275755/>



寫下你的想法：

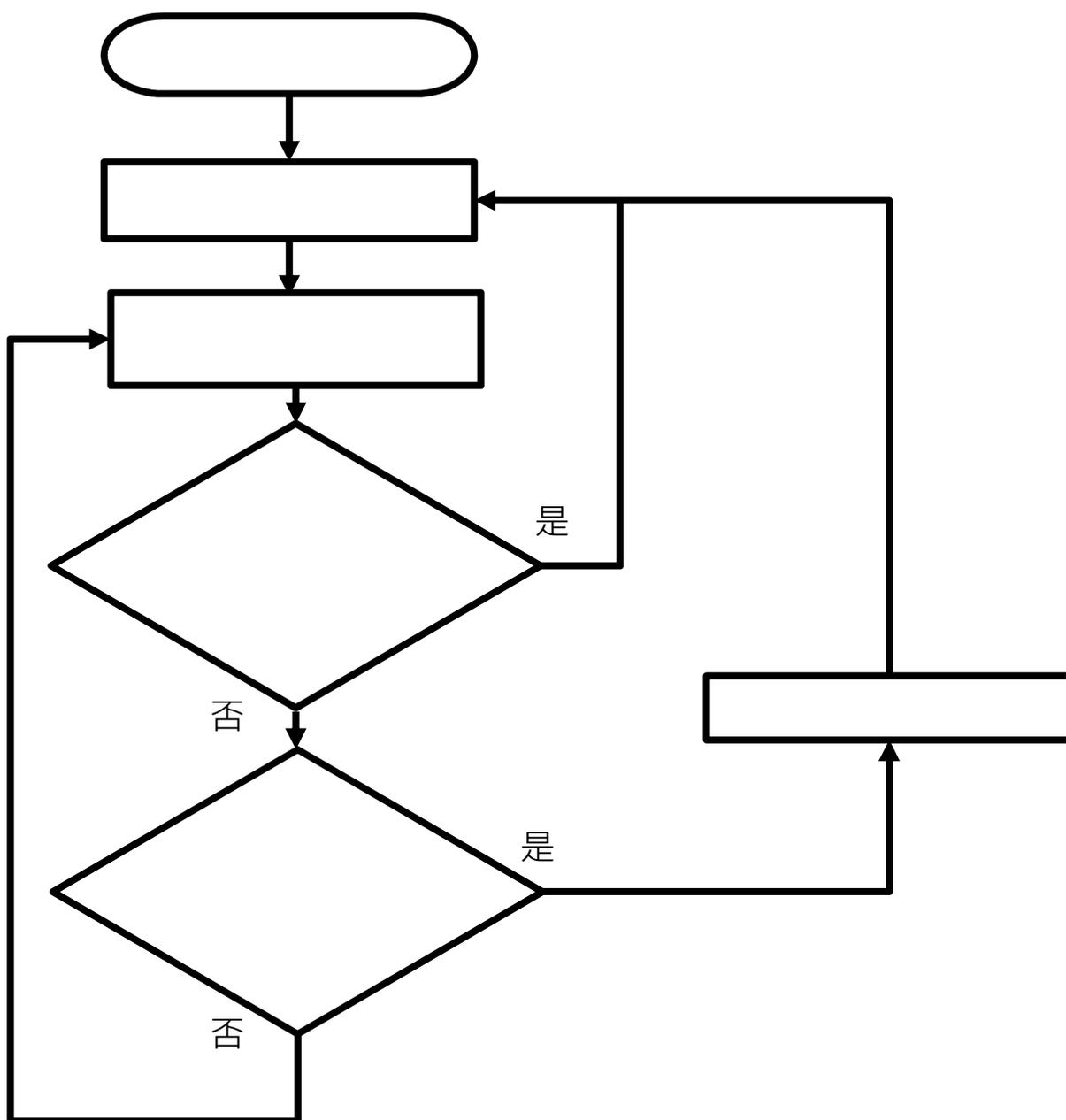
(例如) 如果男孩碰到香蕉  就回到起點。

1. 如果男孩碰到_____，就_____。
2. 如果男孩碰到_____，就_____。

創作迷宮遊戲

想一想、來編程

根據你在本單元中學到的知識，思考自行創作的迷宮遊戲的序列和算法設計。完成流程圖。



運用你設計的流程圖，編程以實現你的想法吧！

創作迷宮遊戲

單元五
學習指引：第四教節

齊反思：「兩顆星星，一個希望」工作紙

專案名稱：_____ 創作者姓名：_____

請寫下在專案中你最喜歡的地方。

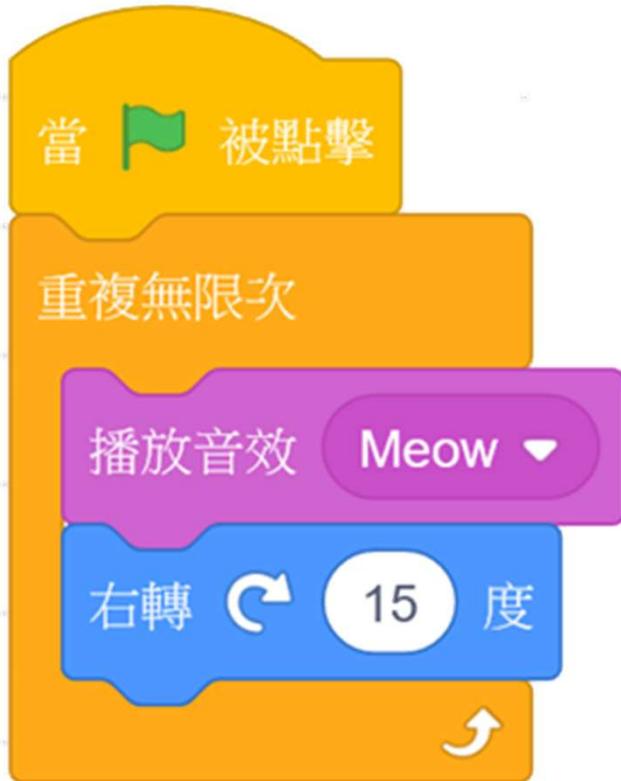


寫出一項你希望為專案加入或更改的東西，使專案變得更好。



重溫練習

1. 當你點擊綠旗時，貓角色會發生什麼？



- A. 貓會發出一聲叫聲，並 15 度轉身一次。
- B. 貓會不斷不停地貓叫，但不會轉身。
- C. 貓會發出一聲叫聲，然後順時針轉身 15 度。
- D. 貓在不斷發出叫聲，然後一直順時針轉身 15 度。

重溫練習

2. 當下面的指令方塊運行時，會發生什麼？



- A. 角色說：「計算器現在是」，然後說「1」、「2」、「3」、「4」、「5」、「6」和「7」，並持續 2 秒。
- B. 角色說：「計算器現在是」，然後說「7」，並持續 2 秒。
- C. 角色說：「計算器現在是」，然後說「counter」。
- D. 角色說：「計算器現在是」，然後說「7」七次，並持續 2 秒。

重溫主要元件

鍵盤事件：



「碰到」指令方塊：

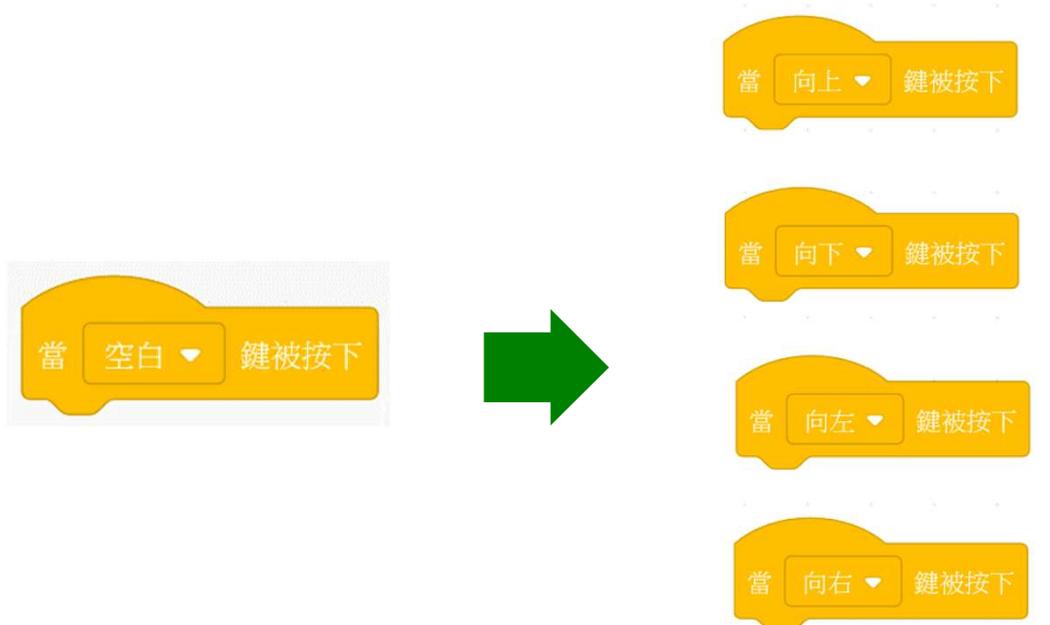


重溫主要概念和實踐

事件：我們用事件指令方塊（鍵盤事件）來觸發 Scratch 做動作（角色移動）。



重用及混合程序 / 編碼：我們在 Scratch 的編程社群，運用重用及混合其他編程人員的作品十分重要。例如，我們可以重用和混合一個鍵盤事件的編碼，應用到其餘的鍵盤事件。



重溫主要概念和實踐

分支 / 選擇：我們在編程使用條件句式進行推理，讓電腦做決定。條件句式總有「如果」的部分，它告訴程式當條件為真成立時，「那麼」應該做什麼。

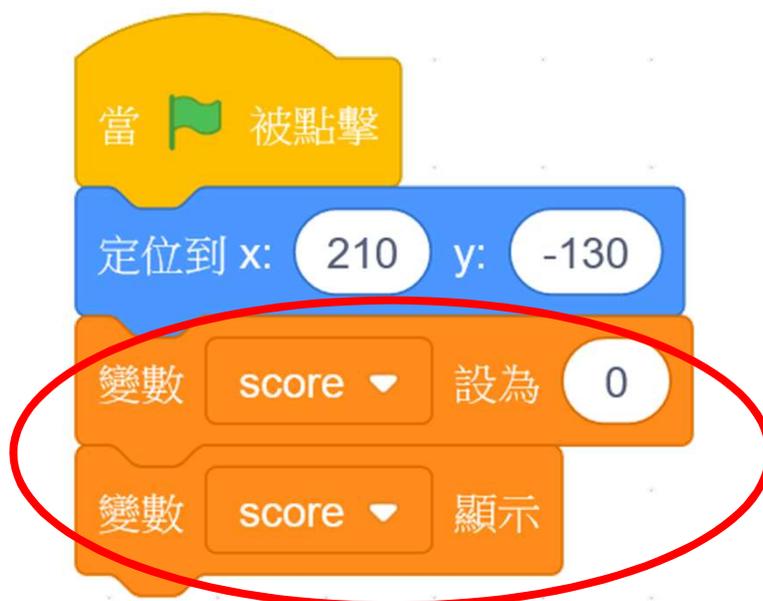


循環 – 重複無限次：循環是一個重覆的過程並藉此產生一系列的輸出。我們可以在 Scratch 運用「重複無限次」(Forever) 引發重覆的動作。



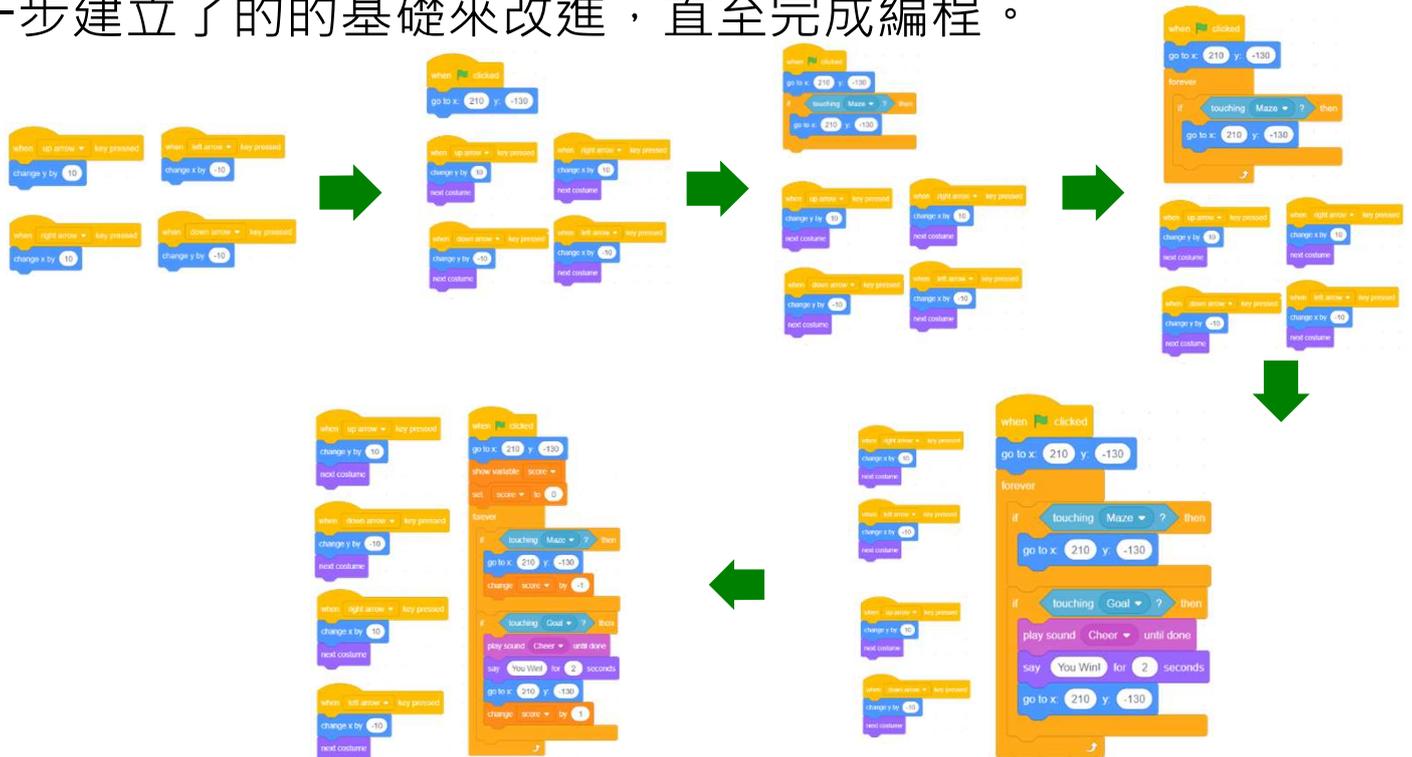
重溫主要概念和實踐

變量：變量在編程用於儲存數值。它有名稱，每次只能儲存一個數值，但可更新該數值。例如，我們建立一個名為「分數」(Score) 的變量來儲存遊戲得分。



重溫主要概念和實踐

反覆構思及漸進編程：這是一種編程技巧。反覆構思即每次計劃一步編程工作，再計劃下一步。漸進意指每次編程都根據上一步建立了的基礎來改進，直至完成編程。



測試及除錯：測試電腦程序是一個檢查它能否按原本的設計進行運作的過程。除錯就是為程序找出錯誤的源頭並改正錯誤。



附錄

操作指南

創作迷宮遊戲

來編程：熊貓在起點

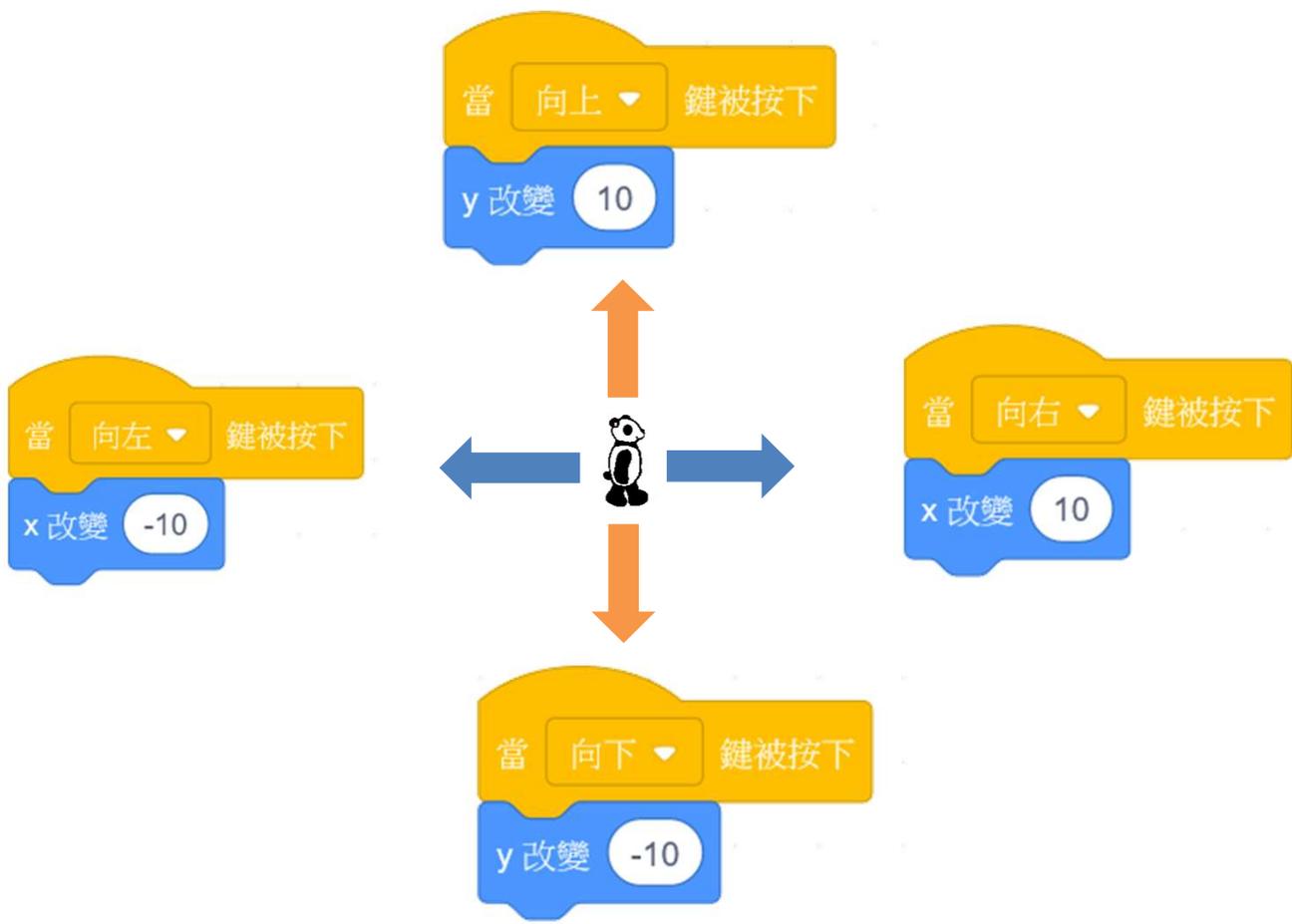
當點擊綠旗時，讓你的角色去到你想要的起始坐標。

見學習指引 P.5



來編程：用鍵盤控制熊貓

見學習指引 P.6

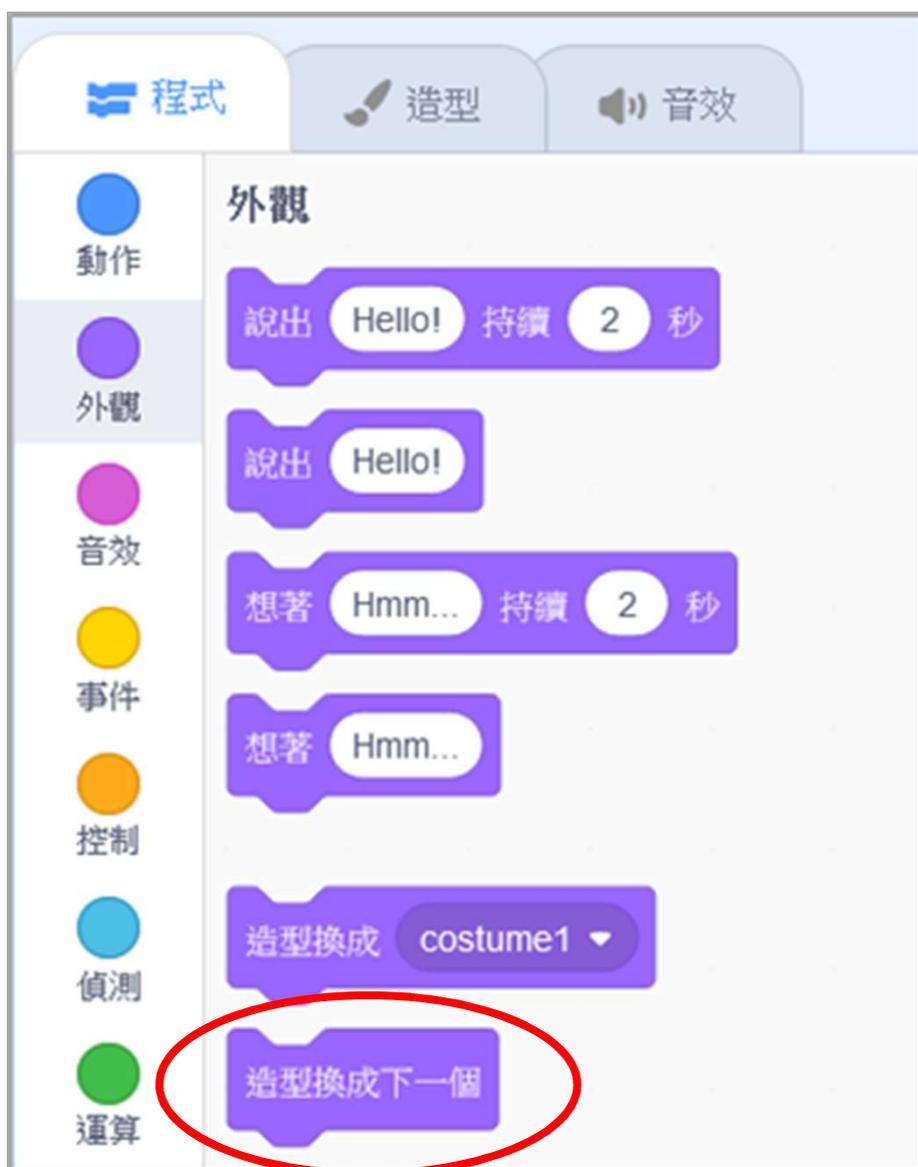
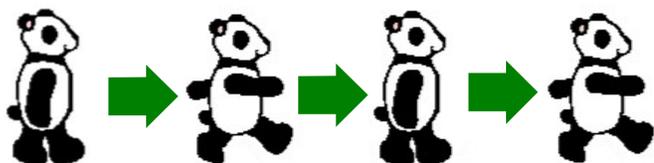


創作迷宮遊戲

想一想、來編程：使熊貓行走

使用「造型換成下一個」方塊讓熊貓看起來更像在走路。

見學習指引 P.7



創作迷宮遊戲

來編程：碰到迷宮

見學習指引
P.12

1. 這個遊戲是熊貓角色碰到迷宮的牆壁，便會送回起點。為達到這目標，請從「控制」欄中拉出一個「如果—那麼」指令方塊來完成這項工作。



2. 我們希望熊貓角色在接觸到 Maze (迷宮) 時回到起點。因此，請從「偵測」欄中拉出一個「碰到 Maze」指令方塊。



3. 將「碰到 Maze」方塊放在「如果—那麼」方塊的空位中，然後將該組合方塊放到「如果綠旗被點擊」方塊的下方。



4. 將「定位到 x:# y:#」方塊複製並貼上到「如果—那麼」指令方塊裏。



創作迷宮遊戲

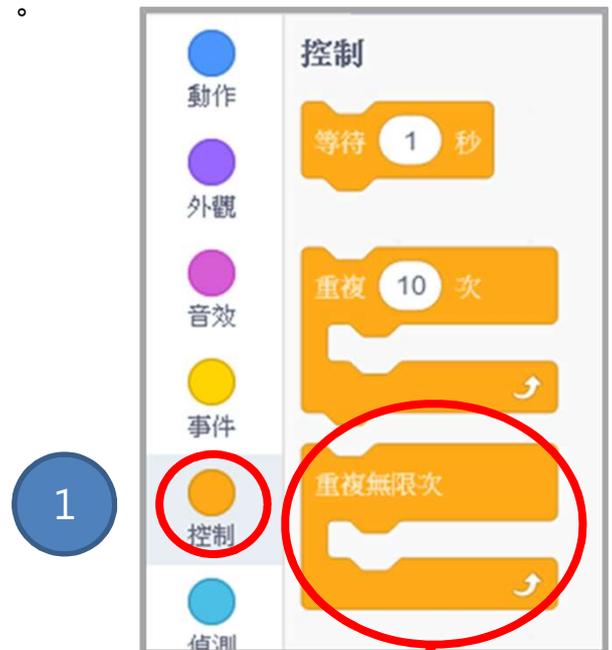
單元五
學習指引：第二教節

想一想、來編程：循環

見學習指引
P.14

「重複無限次」方塊允許程式不斷檢查你正在測試的條件。

1. 從「控制」欄中加入「重複無限次」方塊。



2. 將「如果—碰到 Maze」方塊放進「重複無限次」方塊中。



創作迷宮遊戲

來編程：碰到目標

見學習指引
P.15

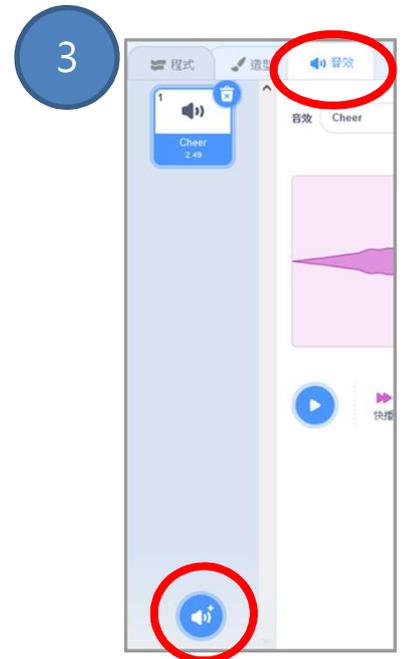
1. 如果熊貓角色「碰到 Goal」（目標竹林）時，則添加附加的「如果-那麼」指令方塊，它就說出“You win！”（你贏了！），並播放聲效。



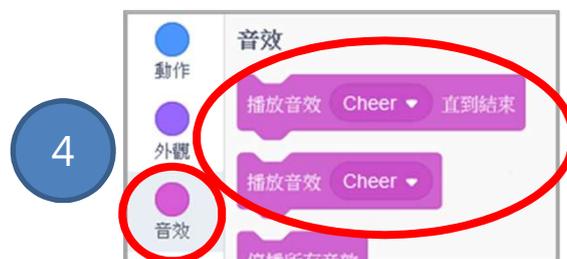
2. 到「外觀」欄選取需要用的方塊。



3. 想要為你的項目添加聲音，請轉到「音效」標籤，點擊左下角的選取音效圖標以選擇聲音。



4. 要讓你的角色播放你找到的聲音，請轉到「程式」標籤，找到「音效」欄。



5. 最後，將熊貓角色送回它起始位置。



創作迷宮遊戲

來編程：加入分數

見學習指引
P.20

1. 把「變數 score 設為 0」和「變數 score 顯示」指令方塊拖到「當綠旗被點擊」下。



2. 當熊貓角色碰到竹林，玩家的分數增加 1 分。拖動「變數 score 改變 1」指令方塊到「如果碰到 Goal」之下。



單元六：運用 micro:bit 創作迷宮遊戲 (延伸單元) 學習指引

目錄

第一教節

玩一玩	S6-1
學一學	S6-3
課前任務 - 學生在家製作操縱桿	S6-4

第二教節

來編程	
加入 micro:bit 擴充功能	S6-11
將 micro:bit 連接到 Scratch	S6-12
想一想、來編程	
用 micro:bit 控制熊貓	S6-13
來編程	
用 micro:bit 控制熊貓	S6-14
循環	S6-16
加入等待指令方塊	S6-17
玩一玩	
迷宮遊戲與數據分析	S6-19
來編程	
程式理解	S6-20

同創作	S6-22
齊反思	S6-23
重溫練習	S6-24
重溫主要元件	S6-25
重溫主要概念和實踐	S6-26

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第一教節

讓我們運用 micro:bit 創作迷宮遊戲來控制角色吧。

這個遊戲跟單元五很相似，但是這次除了鍵盤，你可用操縱桿（內附 micro:bit）控制熊貓。



玩一玩

準備好你的電腦，確保你的 **Scratch Link** 已安裝並正常運行，同時**藍牙**已開啓。
(教師會給你一個帶有 micro:bit 的操縱桿來玩)。



1. 請打開 <https://scratch.mit.edu/projects/734787236/> 並點擊「**切換到程式頁面**」。



2. 點擊橙色的感嘆號並**連接**你的 **micro:bit** 到 Scratch。



3. 用**操縱桿**來玩這個迷宮遊戲，是不是更容易控制並找到竹林呢？



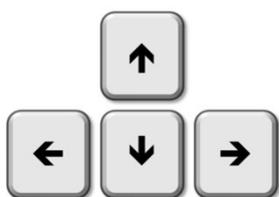
運用 micro:bit 創作迷宮遊戲

想一想

迷宮遊戲在這個單元看起來有些不一樣。完成以下部份標註不同之處。

1. 這次你將**如何控制**熊貓呢？（你可以 多於一個方格）

鍵盤



Micro:bit



滑鼠

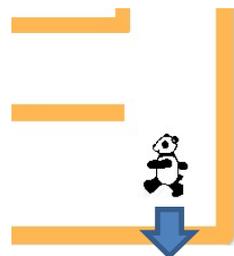


2. 當操縱桿向以下方向**移動**時，熊貓會有什麼**反應**？嘗試匹配下列感測和反應的圖片。

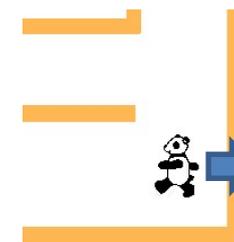
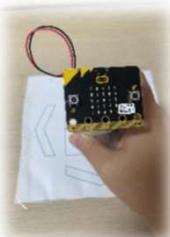
感測

反應

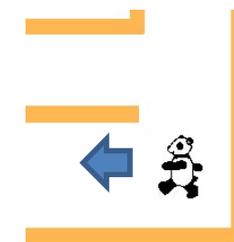
向前



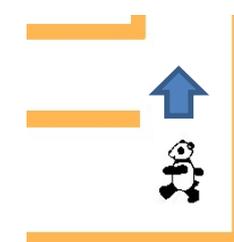
向後



向右



向右

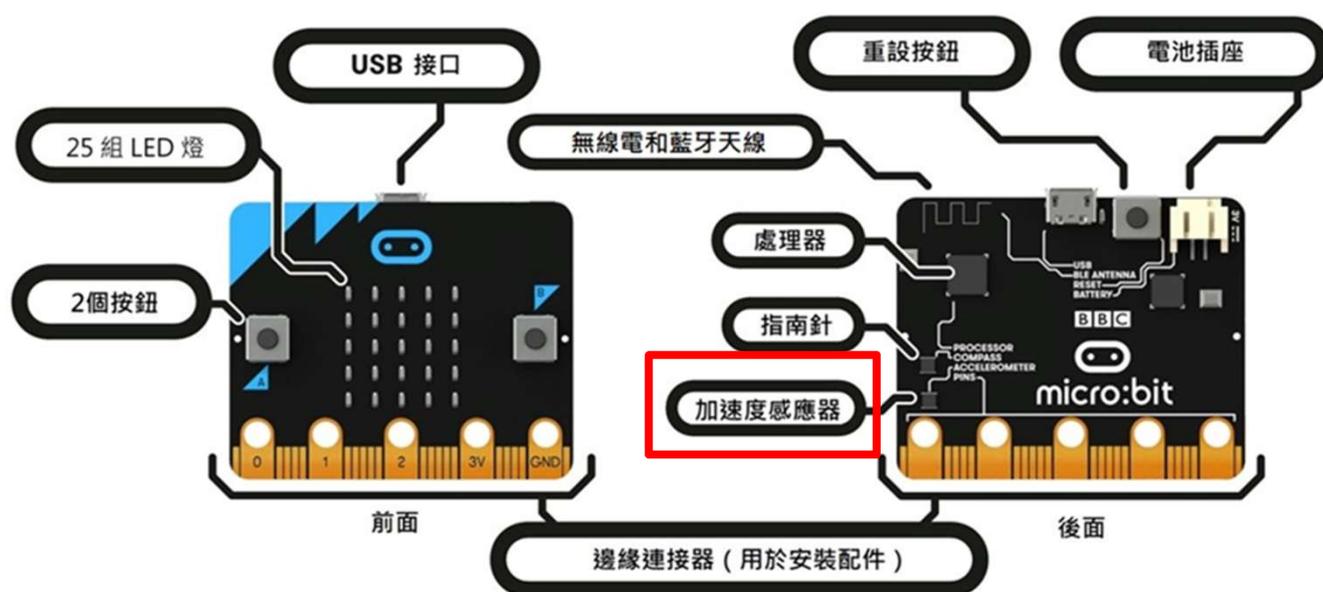


運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第一教節

學一學：micro:bit

我們在這裏可以看到 micro:bit 的所有功能。包括 1 個 LED 顯示屏、2 個按鈕，和 1 個動態感應器（加速度感應器）。你們可以將它連接到 Scratch 並創建一個連結數碼和現實世界的專案。



我們在這個單元，將利用 micro:bit 的功能 – 加速度感應器，結合編程創建一個用操縱桿控制熊貓的遊戲。



知識建立：物聯網（感測—推理—反應）



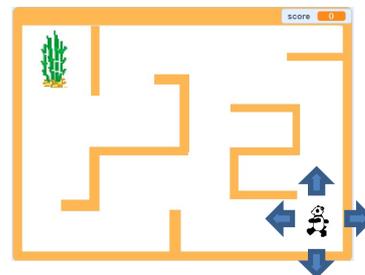
感測

從感應器收集數據



推理

根據數據作出判斷和決定



反應

根據推理結果作出回應

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第一教節

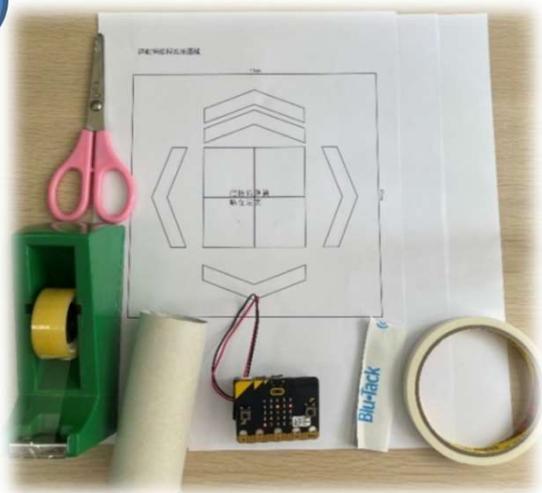
課前任務（學生在家製作操縱桿）

你可以按照以下步驟嘗試製作一個操縱桿模型：

1. 準備以下手工材料：

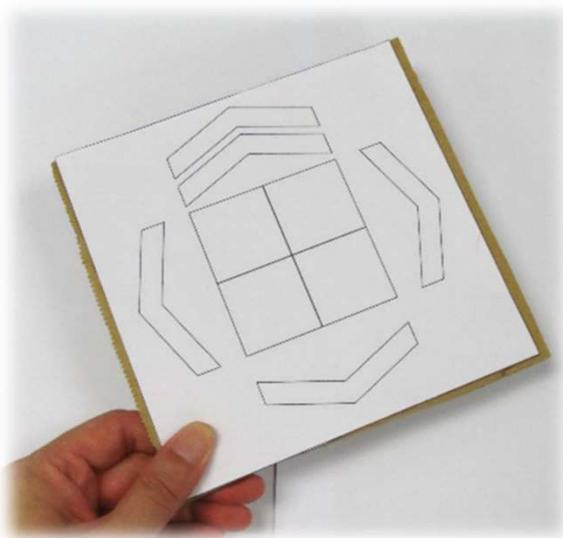
- 已插電的 micro:bit
- 廁紙筒
- 2 張 A4 紙
- 操縱桿圖紙
- 透明膠紙
- 雙面膠紙
- 剪刀
- 寶貼萬用膠（Blu-Tack）

1



- ### 2. 打印並裁剪出位於下一頁的操縱桿圖紙（你可以將它貼在相同大小的硬紙板上，或者在上課時將它貼在桌子上）。

2

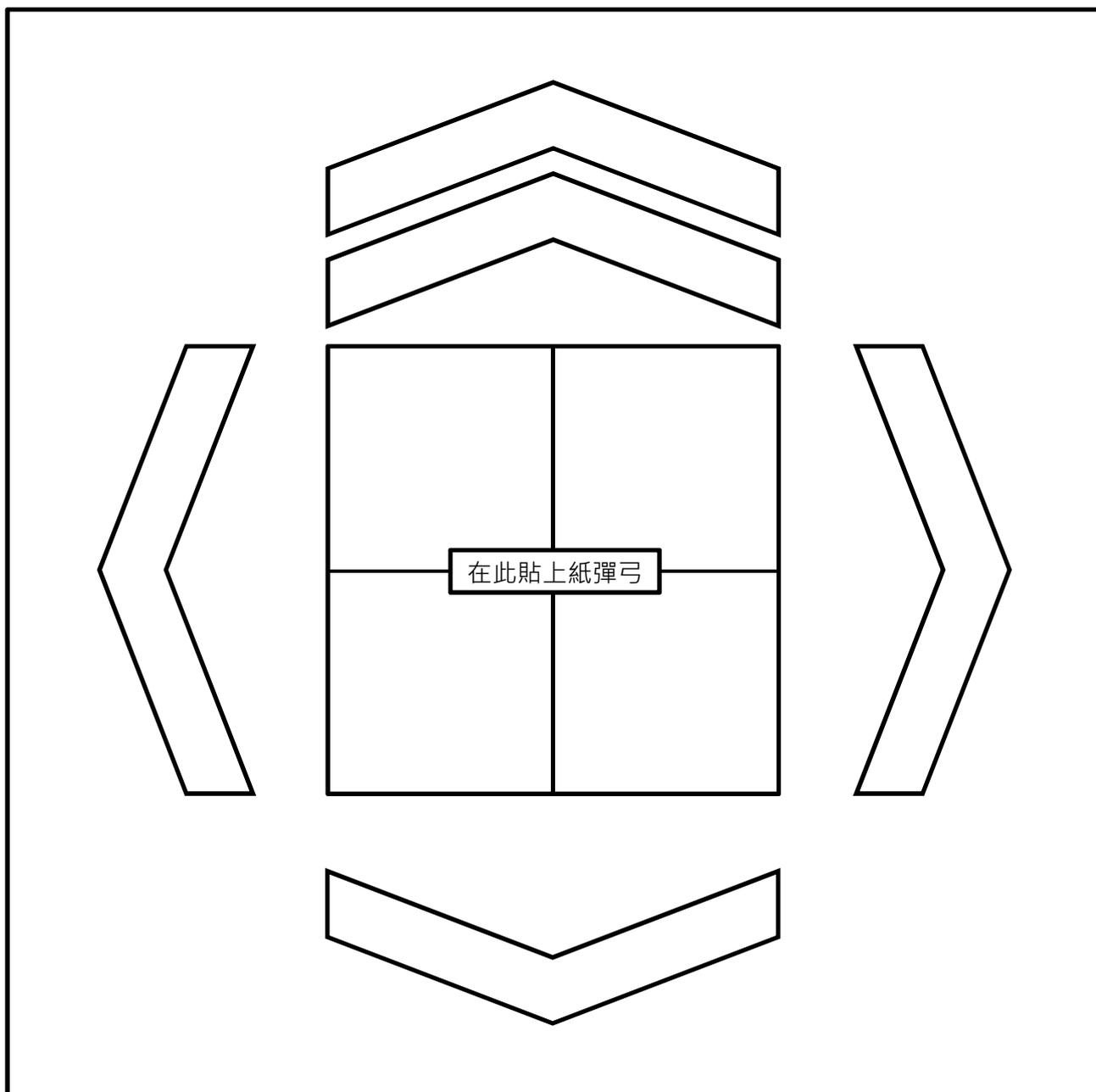


運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第一教節

課前任務（學生在家製作操縱桿）

操縱桿圖紙：



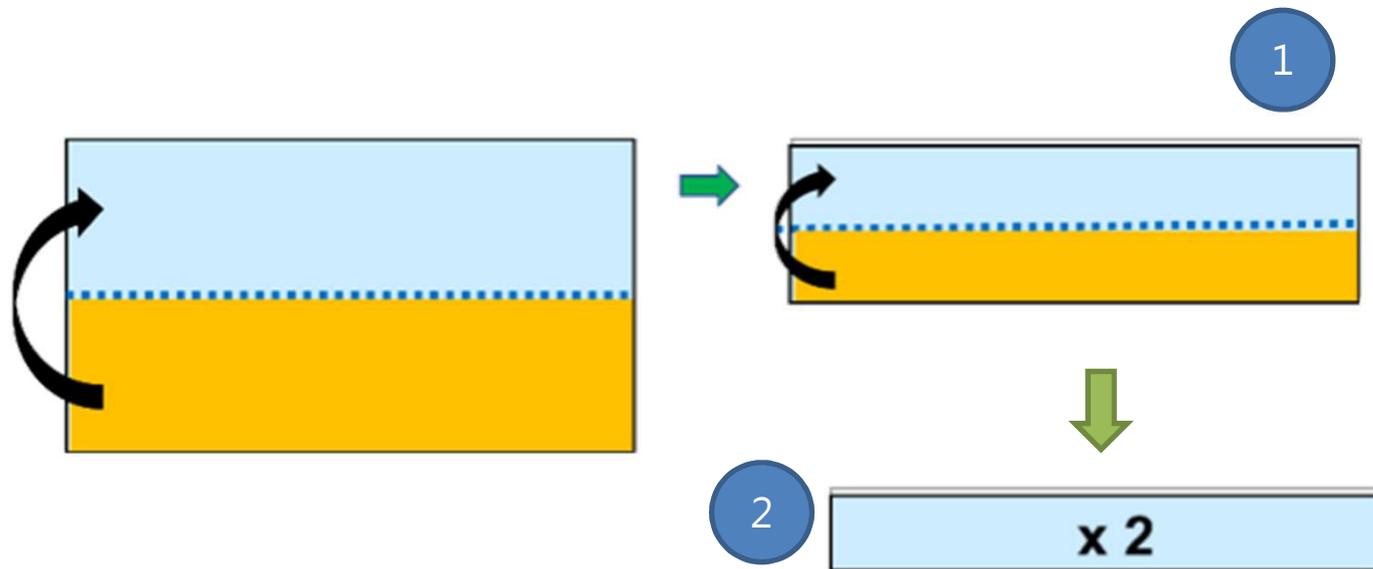
空白頁面

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第一教節

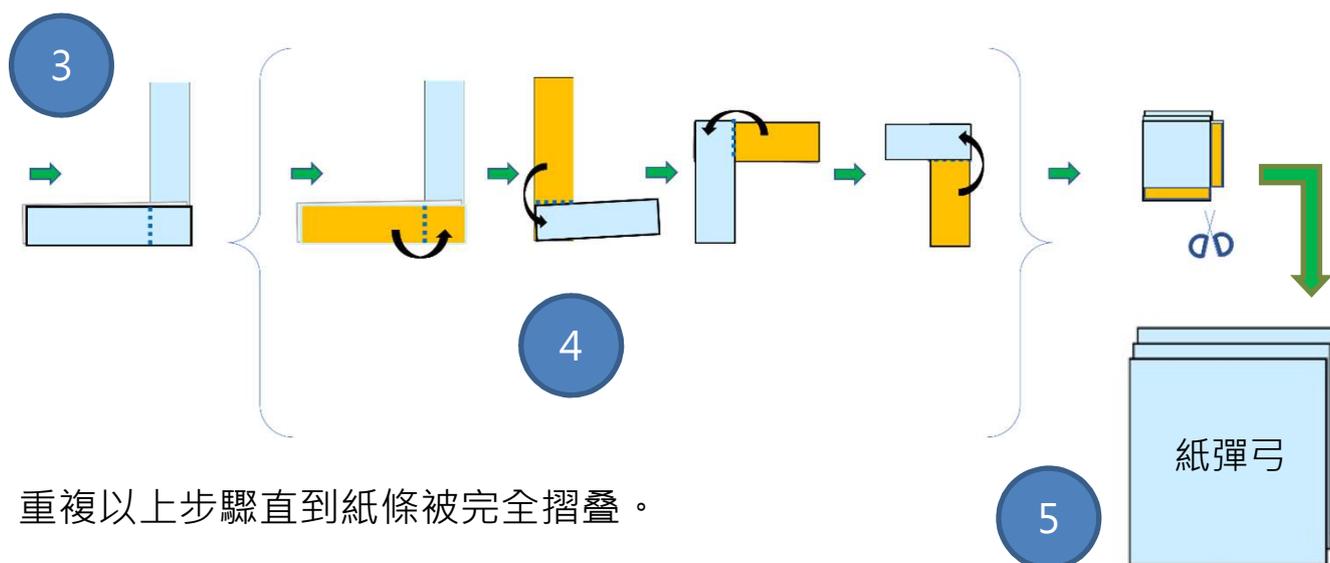
課前任務（學生在家製作操縱桿）

1. 用 A4 紙摺出一條紙彈弓。



2. 將 2 張 A4 紙垂直對摺兩次，摺出 2 個厚紙條。

3. 左手水平拿住一個紙條，右手將另一紙條垂直固定在水平紙條的末端。



4. 重複以上步驟直到紙條被完全摺疊。

5. 剪掉多餘部分，用雙面膠紙固定紙條兩端，形成一個紙彈弓。

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第一教節

課前任務（學生在家製作操縱桿）

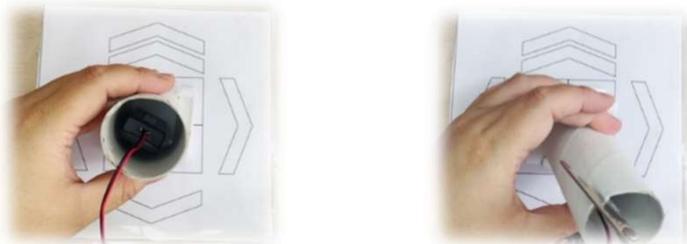
1. 用剪刀在廁紙筒一端剪出 4 個長約 1 cm 的缺口；向外摺疊這些缺口，用雙面膠紙將它們貼在紙彈弓。



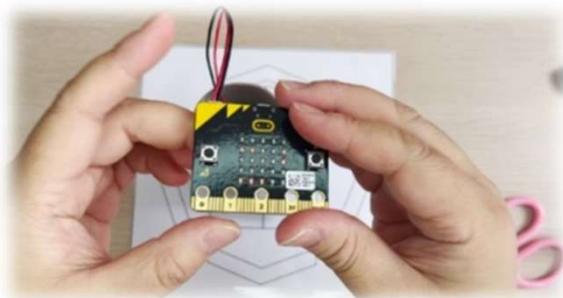
2. 用雙面膠紙將紙彈弓的另一端固定在草圖紙的中心。



3. 當你從教師那拿到 micro:bit（或者你可以在課堂上完成最後幾步），把 micro:bit 的電池盒放進廁紙筒，並剪出 4cm 的縫隙隱藏電線；然後，用 1 個寶貼萬用膠（Blu-Tack）將 micro:bit 貼在廁紙筒頂部。

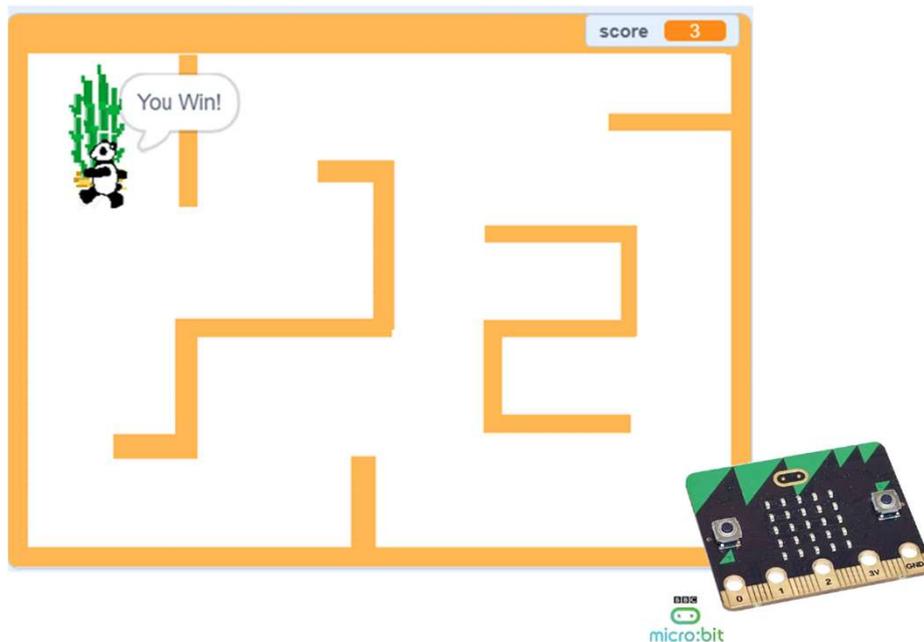


4. 操縱桿製作完成！



運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節



從這裏開始

1. 請同學在 scratch.mit.edu 登錄你的賬號。

1

2. 請打開上一個單元已經完成的迷宮遊戲：

<https://scratch.mit.edu/projects/722154863>

作為本單元的範例，並點擊「改編」按鈕。

2

改編

切換到程式頁面



3. 重新命名專案為 **Maze Game with microbit**。

3



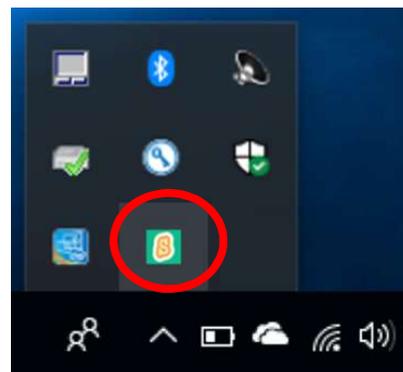
運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

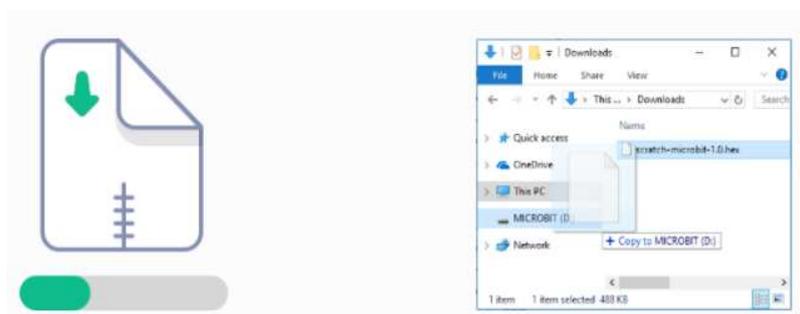
準備工作

1. 若將 micro:bit 連接到 Scratch，需要下載並安裝 Scratch Link 軟件。你可以在此連結中取得全部所需資源：<https://scratch.mit.edu/microbit>

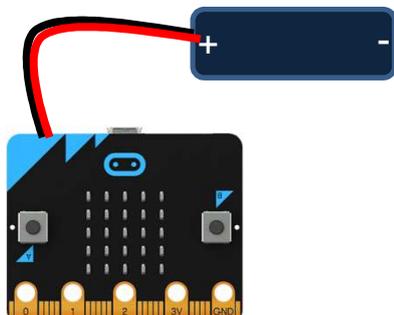
2. 啓動 Scratch Link 並確保它在運行中。
它會在工具欄中出現。



3. 請下載 Scratch micro:bit HEX 文件，並將其拖動至 micro:bit 磁碟中。



4. 用 USB 線 或電池盒，讓 micro:bit 接上電源。



運用 micro:bit 創作迷宮遊戲

來編程：加入 micro:bit 擴充功能

現在回到你的 Scratch 項目。

1. 點擊在 Scratch 編程界面左下角的「添加擴展」按鈕。



2. 點擊 micro:bit 擴充功能。

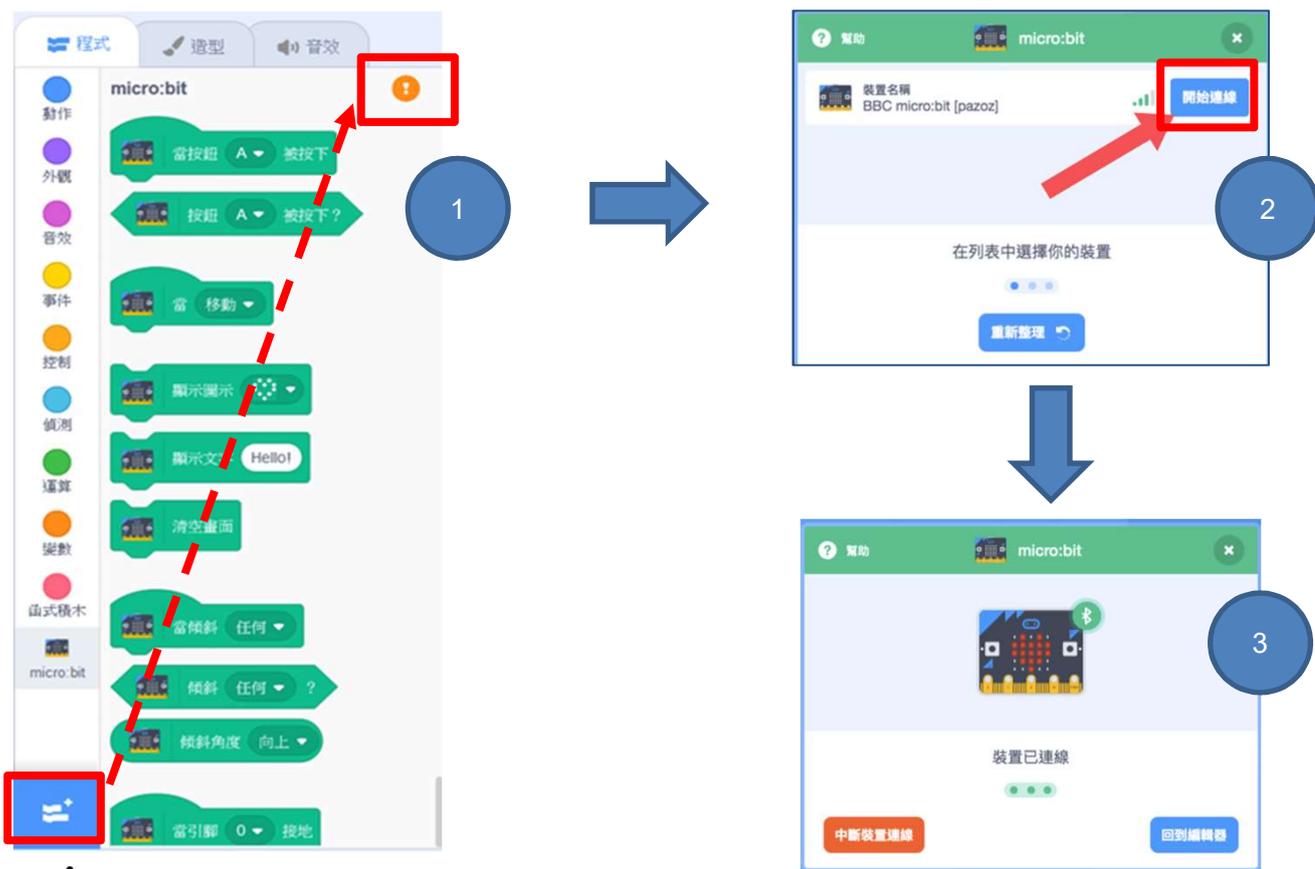


運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

來編程：將 micro:bit 連接到 Scratch

1. 請在「程式」標籤頁的“micro:bit”組別，點擊圖標（橙色  感嘆號）。
2. 請在相應設備選項上點擊「開始連線」按鈕連接 micro:bit。
3. 點擊「回到編輯器」按鈕並返回編程平台。



知識建立：因果推理

透過尋找因果關係的過程來解決問題，探討不同源頭產生的效果，找出問題的解決方法。如果你還沒有做好所需準備，或缺少某些東西。

例如：藍牙未開啓，micro:bit 電量不足，你可能無法將 micro:bit 連接到 Scratch（如圖所示）。



運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

想一想、來編程：用 micro:bit 控制熊貓

試比較上個單元我們完成的專案，思考需改變哪一部分的程式指令？



為了用操縱桿控制熊貓而不是鍵盤，我們應怎樣更新指令方塊？

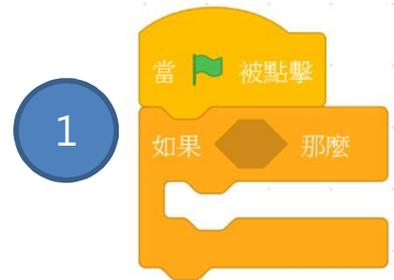
提示：查看 micro:bit 欄。



運用 micro:bit 創作迷宮遊戲

來編程：用 micro:bit 控制熊貓

1. 拖出一個「當綠旗被點擊」和一個「如果-那麼」指令方塊，並將他們組合起來。



2. 從新添加的 micro:bit 選項中，找到「傾斜 <任何>?」指令方塊。
3. 將「任何」改為「向上」，並將其插入「如果」指令方塊中。
4. 複製角色移動等的指令方塊到「當鍵被按下」事件指令方塊。

micro:bit

- 當按鈕 A 被按下
- 按鈕 A 被按下?
- 當 移動
- 顯示圖示
- 顯示文字 Hello!
- 清空畫面
- 當傾斜 任何
- 傾斜 任何?
- 傾斜角度 向上

當 綠旗 被點擊

如果 傾斜 向上? 那麼

x 改變 10

造型換成下一個

當 向右 鍵被按下

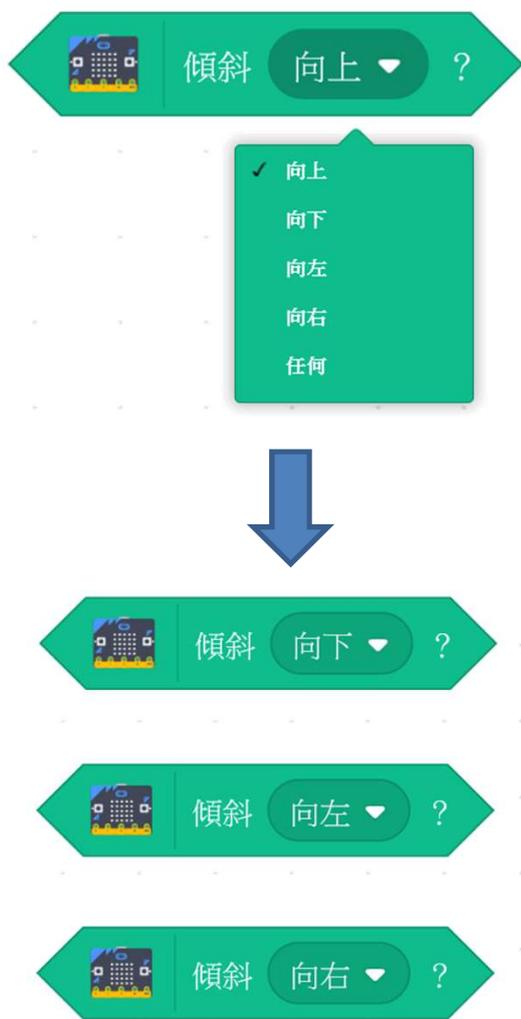
x 改變 10

造型換成下一個

運用 micro:bit 創作迷宮遊戲

來編程：用 micro:bit 控制熊貓

複製「傾斜向上」指令方塊，更新並完成其他3個方向的剩餘部分：



測試及除錯

你能用 micro:bit 控制角色嗎？
能試多幾次嗎？欠缺了哪個指令方塊？



運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

來編程：循環

你會運用哪個指令方塊來**解決**問題？

提示：查看在「**控制**」欄。



測試及除錯

再試一次吧！操縱桿是否太敏感呢？
怎麼解決這個問題呢？

用 micro:bit 控
制熊貓



循環



運用 micro:bit 創作迷宮遊戲

來編程：加入等待指令方塊

操縱桿會否太敏感？你又會怎樣解決？

加入「等待__秒」指令方塊看看能否幫助你。



我們應等待多久？嘗試不同數值，例如 0.3 秒、1 秒或 3 秒直至你找到合適的時間。



測試及除錯

你做到了！快用 micro:bit 操縱桿感受迷宮遊戲的樂趣吧！



用 micro:bit 控制熊貓

循環

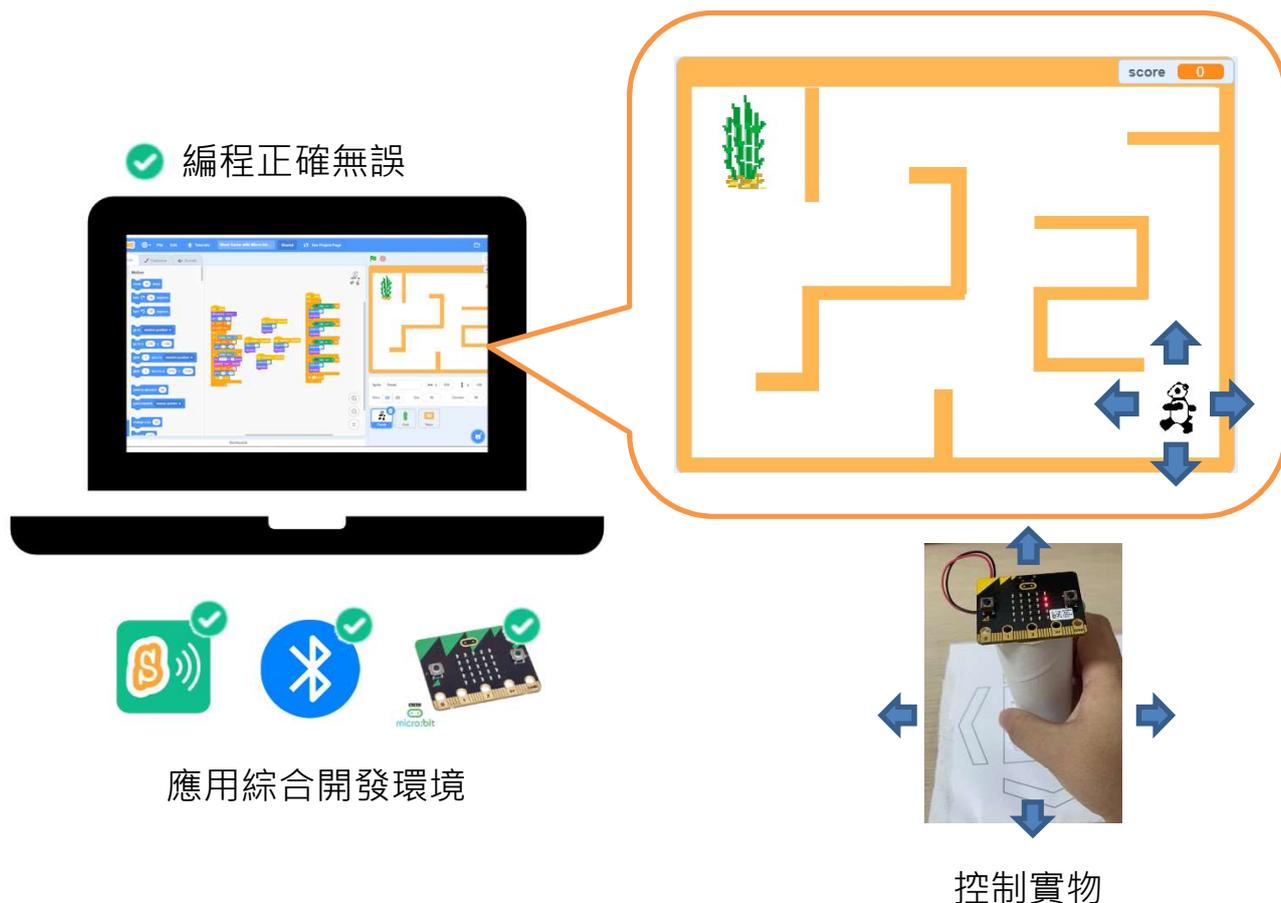
加入等待指令方塊

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

知識建立：工程系統思維 / 連接實物建構系統

我們須準備好環境並接通所有組成部分，否則將無法啟動遊戲 / 系統。



知識建立：工程系統思維 / 連接實物建構系統

將相互關聯的部分作為一個整體組合在一起，並使實物系統按預期運作的思維。

運用 micro:bit 創作迷宮遊戲

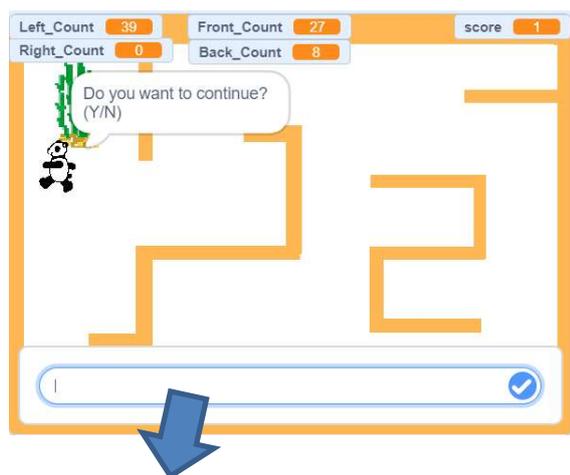
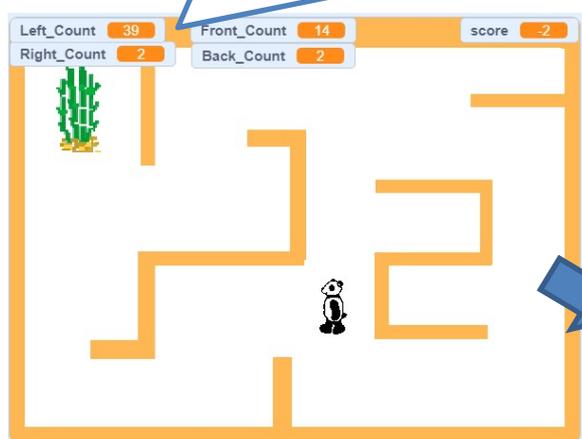
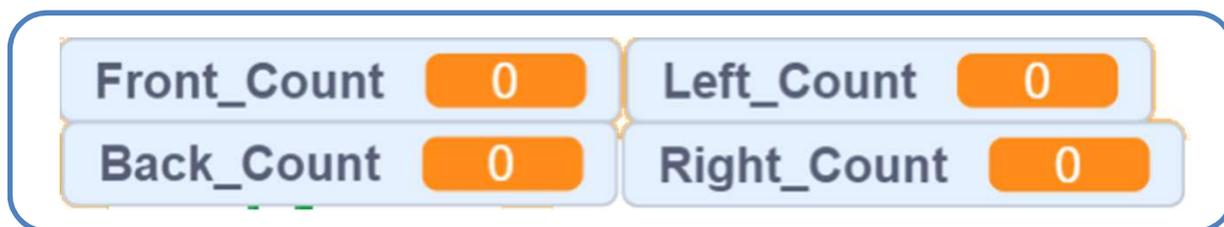
單元六
學習指引：第二教節

玩一玩：迷宮遊戲與數據分析

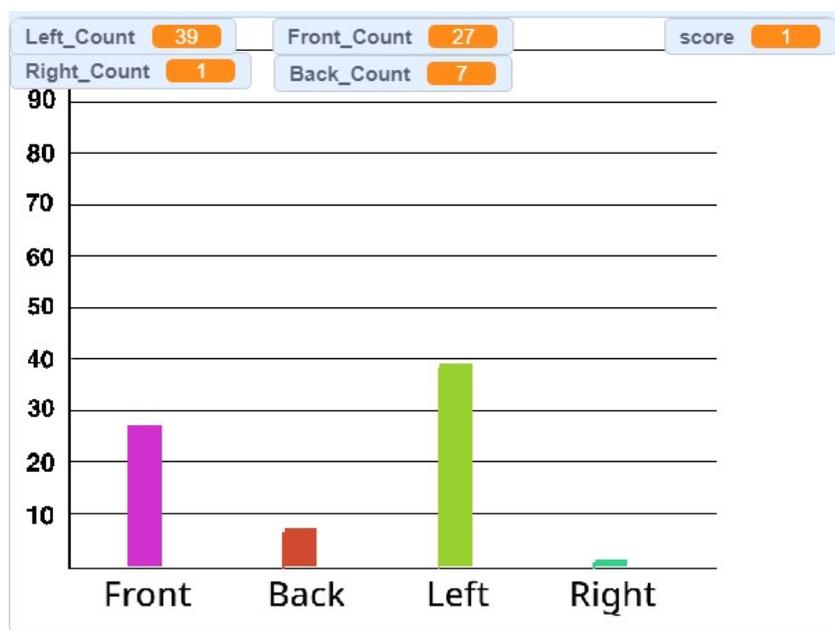
現在來玩帶有數據分析的迷宮遊戲吧，它會計算你的步數並以條形圖的形式顯示。

<https://scratch.mit.edu/projects/734745981/>

除了得分，你還會看到四個變數用來計算角色前 / 後 / 左 / 右的移動。



輸入 "N" 完成遊戲，你將會看到你步數的條形圖。



運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

來編程：程式理解

點擊「[切換到程式頁面](#)」查看程式。

1. 我們有四個變數計算你的步數。它包括用不同的變數計算你的步數並在圖表中顯示數據。
2. 無論你是否用鍵盤或 micro:bit 控制，它都會計算你的移動。

```
當 向上 鍵被按下
變數 Front_Count 改變 1
y 改變 10
造型換成下一個

當 被點擊
重複無限次
如果 傾斜 向上 ? 那麼
變數 Front_Count 改變 1
y 改變 10
造型換成下一個
```

變數

建立一個變數

- Back_Count
- barWidth
- Front_Count
- Left_Count
- Right_Count
- score

3. 當你完成這個遊戲，它會「[詢問](#)」你是否要繼續。

```
如果 碰到 Goal ? 那麼
說出 You Win! 持續 2 秒
播放音效 Cheer 直到結束
變數 score 改變 1
詢問 Do you want to continue? (Y/N) 並等待
如果 詢問的答案 = Y 那麼
定位到 x: 210 y: -130
等待 1 秒
否則
廣播訊息 displayStatistics
```



4

4. 通過使用「[畫筆](#)」功能你會發現這兒有一個「Point」（角色）會在遊戲期間用計算出的數據繪製條形圖。

```
當收到訊息 displayStatistics
顯示
筆跡全部清除
筆跡寬度設為 1
筆跡顏色設為 藍色
筆跡 顏色 改變 20
變數 barWidth 設為 20
變數 startX 設為 -172
變數 startY 設為 -140
定位到 x: startX y: startY
重複 barWidth 次
下筆
y 改變 Front_Count * 3
停筆
x 改變 1
y 設為 startY
變數 startX 設為 startX + 90
筆跡 顏色 改變 20
```

總結

本單元是一個簡化版的物聯網（感測 – 推理 – 反應）。配合操縱桿（內置 micro:bit），我們運用 micro:bit 內置的感測器（加速度感應器），編程並與 Scratch 編程環境互動，從而形成一個與實物連接的系統。

延伸概念：物聯網 Internet Of Things (IoT)

物聯網 (IoT)：它是由感測器、應用程式及其他相關技術嵌入而成的實物網絡，目的在於透過互聯網與其他裝置設備交換數據。它通常使用的通訊技術有藍牙和無線網絡 (Wi-Fi)。

透過運用物聯網，我們可以協調學生在上學日的學習和活動。

你還知道生活中使用物聯網的例子嗎？

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

同創作

你想運用今天學到關於 micro:bit 的加速度感應器創造什麼呢？請在下列方框內畫出你的想法：



運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

齊反思：「兩顆星星，一個希望」工作紙

專案名稱：_____ 創作者姓名：_____

請寫下在專案中你最喜歡的地方。



寫出一項你希望為專案加入或更改的東西，使專案變得更好。



運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

重溫練習

1. 一名學生嘗試用 micro:bit 控制角色，但未能移動到正確的方向。理解以下指令方塊並嘗試除錯。



- A. 移除 重複無限次 指令方塊。
- B. 將等待秒數由 "0.2" 調整為 "2" 。
- C. 將第四個 如果-那麼 條件句式中的「傾斜任何」改為「傾斜向右」。
- D. 改用另一個 micro:bit 。

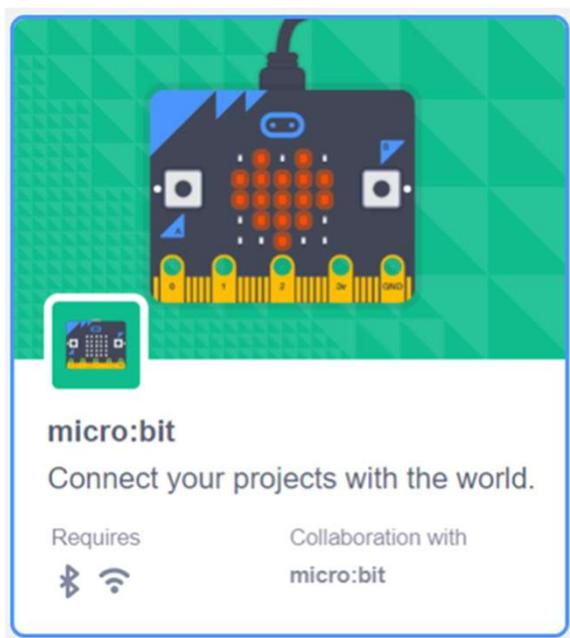
運用 micro:bit 創作迷宮遊戲

重溫主要元件

micro:bit 和加速度感應器：



運



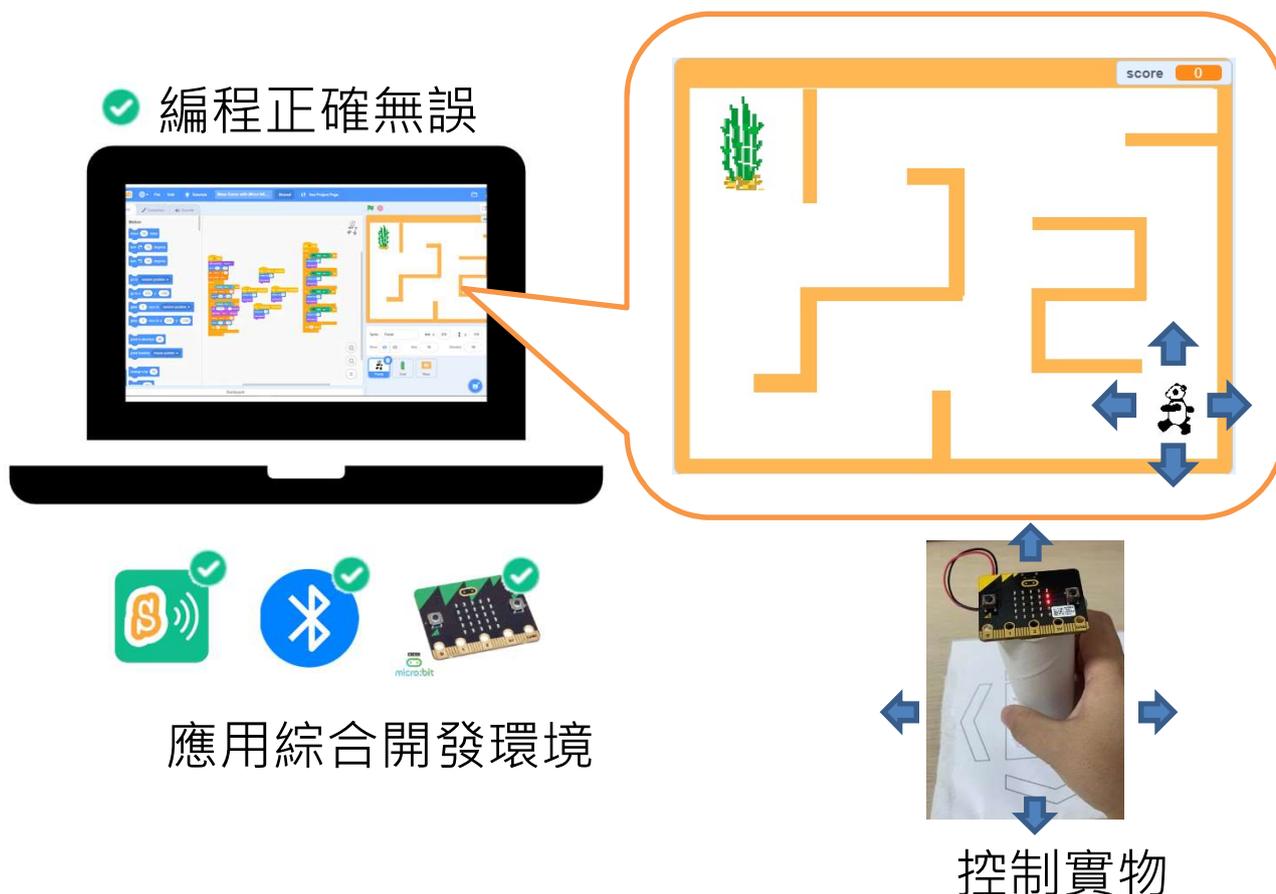
重溫主要概念和實踐

感測：從感應器收集數據
推理：根據數據作出判斷和決定
反應：根據推理結果作出回應



工程系統思維 / 連接實物建構系統：

將相互關聯的部分作為一個整體組合在一起，並使實物系統按預期運作的思維。

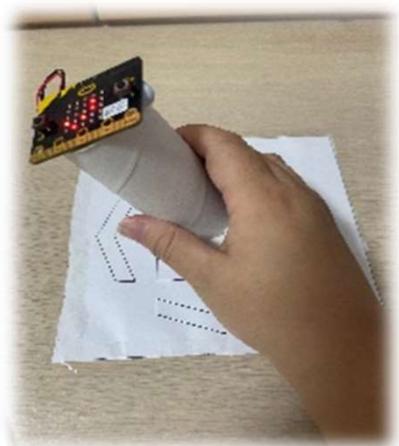
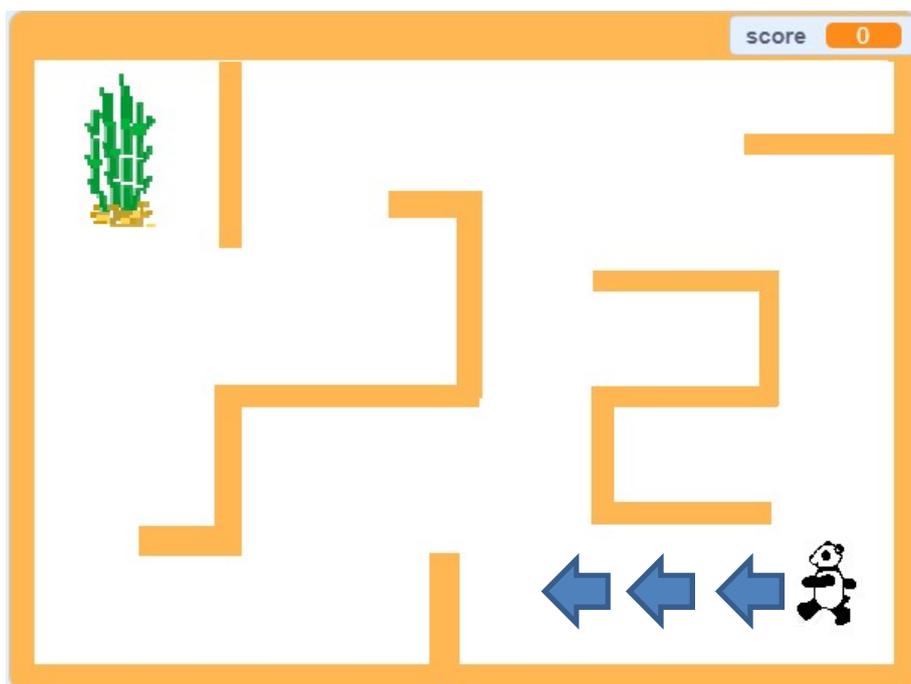


運用 micro:bit 創作迷宮遊戲

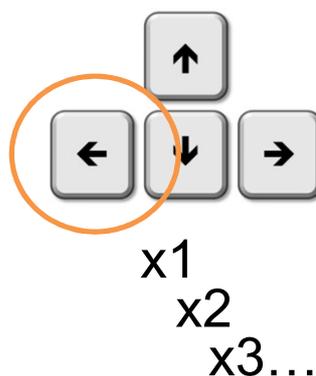
單元六
學習指引：第二教節

重溫主要概念和實踐

運用 micro:bit 的加速度感應器操縱桿取代鍵盤操縱角色



保持移動



一鍵，一步

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

重溫主要概念和實踐

排序：按特定順序組合事物的過程。



因果關係：透過尋找因果關係的過程來解決問題，探討不同源頭產生的效果，找出問題的解決方法。



錯誤畫面



檢查項目

運用 micro:bit 創作迷宮遊戲

單元六
學習指引：第二教節

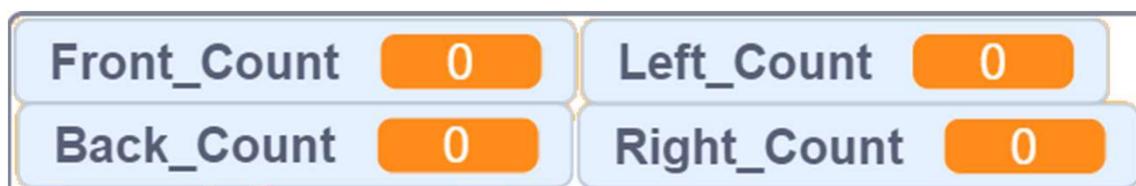
重溫主要概念和實踐

分支 / 選擇：我們在編程使用條件句式進行推理，讓電腦做決定。條件句式總有「如果」的部分，它告訴程式當條件為真成立時，「那麼」應該做什麼。

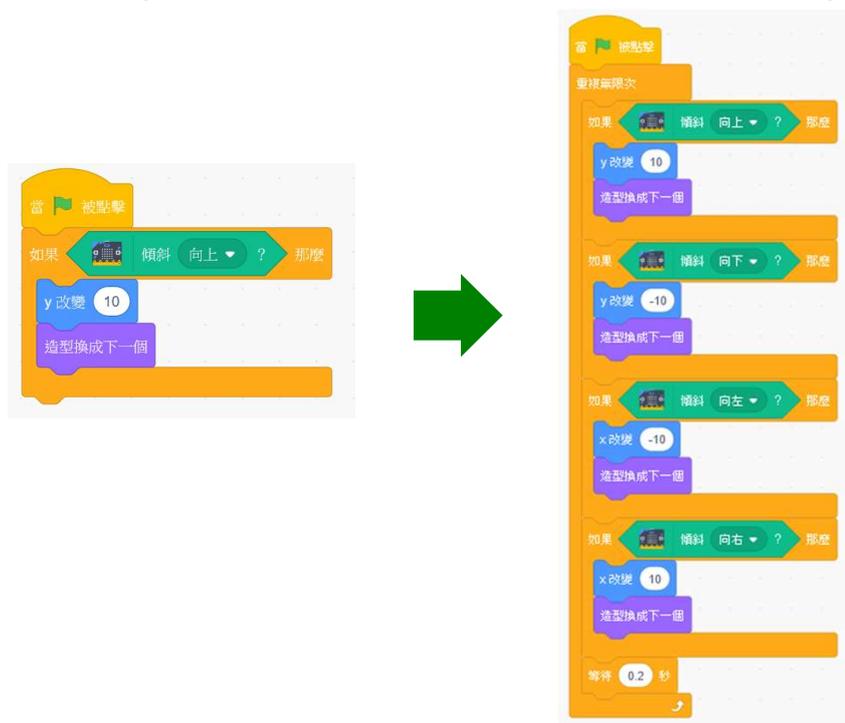


重溫主要概念和實踐

變量：變量在編程用於儲存數值。它有名稱，每次只能儲存一個數值，但可更新該數值。例如，我們建立一個名為「分數」(Score) 的變量來儲存遊戲得分。

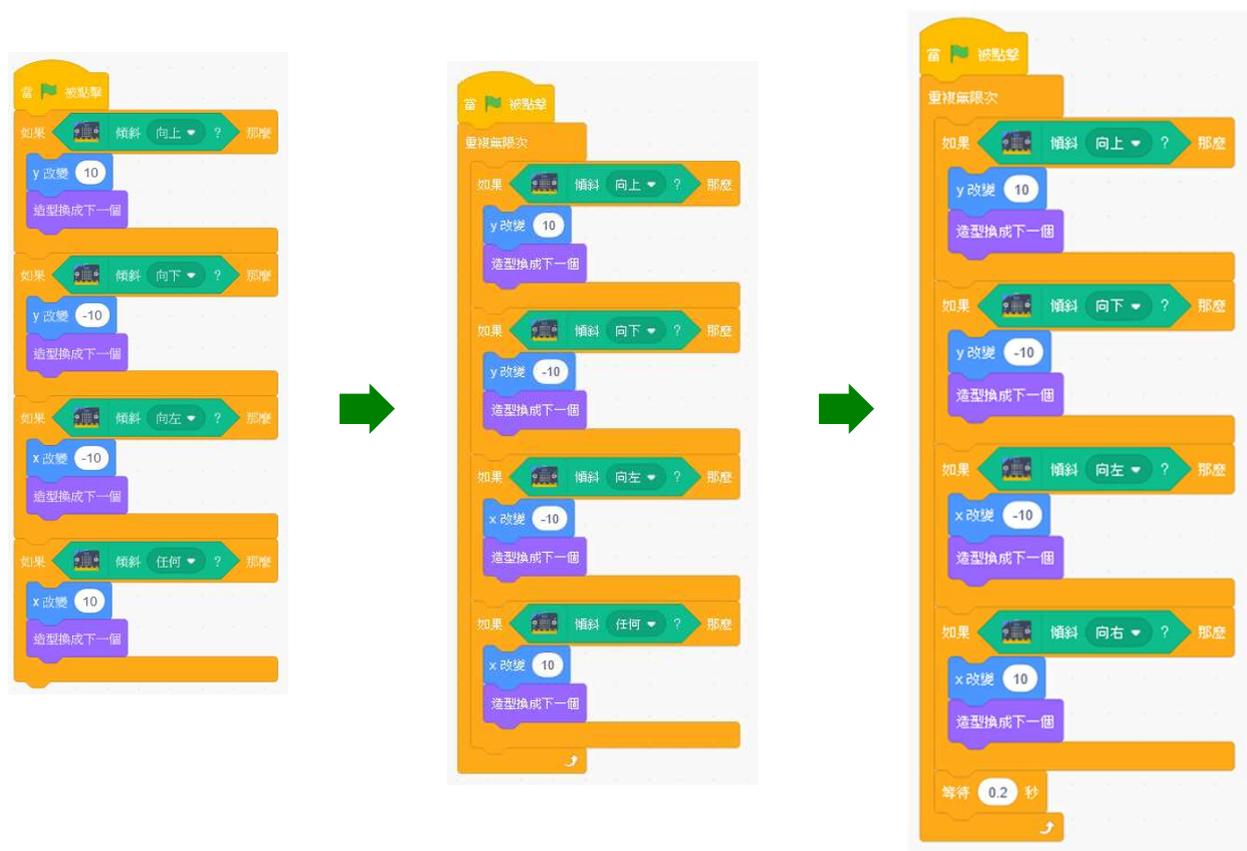


重用及混合程序 / 編碼：我們在 Scratch 的編程社群，運用重用及混合其他編程人員的作品十分重要。我們可以重用和混合一個事件（例如「傾斜向上」）的編碼，應用到其他事件（例如「傾斜向下 / 左 / 右」）。



重溫主要概念和實踐

反覆構思及漸進編程：這是一種編程技巧。反覆構思即每次計劃一步編程工作，再計劃下一步。漸進意指每次編程都根據上一步建立了的基礎來改進，直至完成編程。



測試及除錯：測試電腦程序是一個檢查它能否按原本的設計進行運作的過程。除錯就是為程序找出錯誤的源頭並改正錯誤。



單元七：運用 Scratch 畫圖形 學習指引

目錄

第一教節

玩一玩	S7-1
想一想	S7-2
來編程	
新增畫筆功能	S7-4
繪畫一條線	S7-5
想一想、來編程	
繪畫正方形	S7-6
使用重複指令方塊	S7-7

第二教節

想一想	
一塊雪花 = 多個正方形	S7-9
想一想、來編程	
繪畫多個正方形	S7-10

第三教節

想一想、來編程	
創建“DrawASquare”指令方塊	S7-15
創建“DrawASnowflake”指令方塊	S7-18
齊反思	S7-19

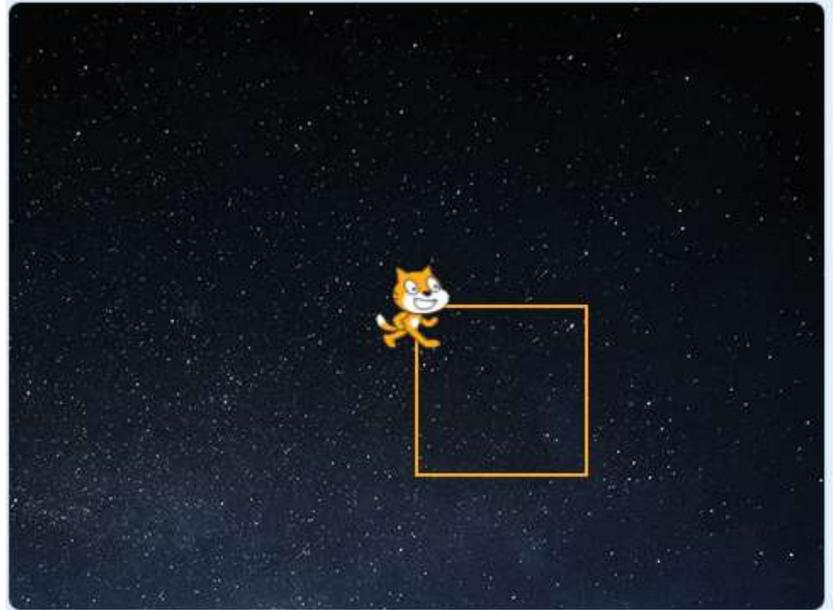
重溫練習	S7-20
重溫主要元件	S7-22
重溫主要概念和實踐	S7-24
附錄 - 操作指南	S7-27

運用 Scratch 畫圖形

單元七
學習指引：第一教節

讓我們一起學習如何在 Scratch 中繪畫圖形吧！

你將在這單元學習如何使用 Scratch 的畫筆功能來繪畫一條線和正方形。



玩一玩

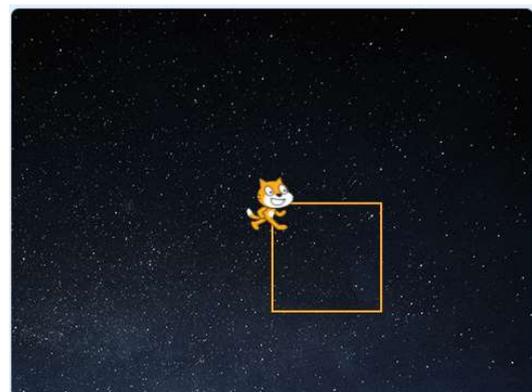
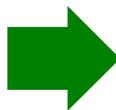
打開 Scratch 專案：運用Scratch畫圖形。

<https://scratch.mit.edu/projects/737338011/>

點擊綠旗看看會發生什麼事情。

再次點擊綠旗，看看**角色移動**到哪裏。

Scratch 小貓畫出一個**正方形**嗎？

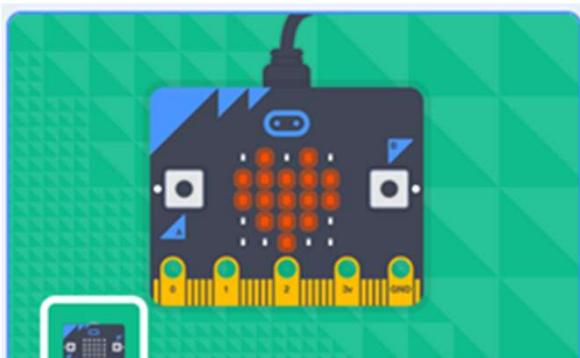


運用 Scratch 畫圖形

想一想

1. 你知道在這單元中我們會使用哪個功能來繪畫一個正方形嗎？請在合適的方格上加上（✓）。

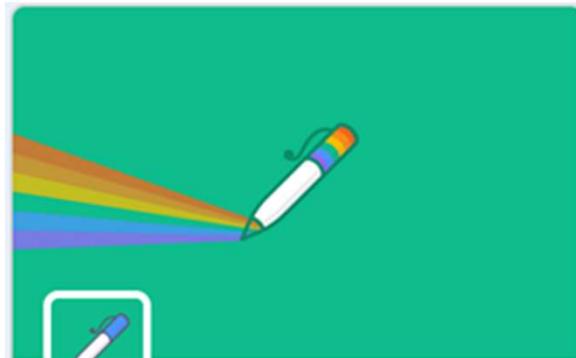
Micro:bit



micro:bit
讓你的專案與實體世界連結。

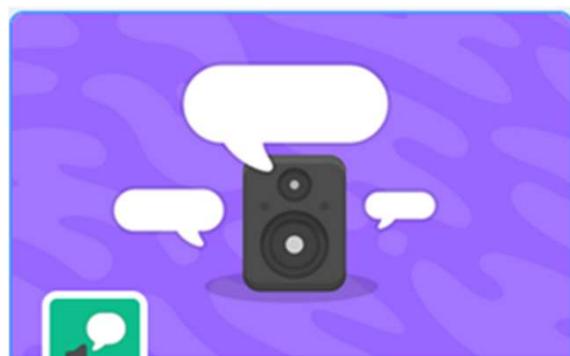
需求   合作者
micro:bit

畫筆



畫筆
使用你的角色來畫圖。

文字轉語音



文字轉語音
讓你的專案能夠說話

需求  合作者
Amazon Web Services

視訊偵測



視訊偵測
使用攝影機偵測動作。

運用 Scratch 畫圖形

從這裏開始

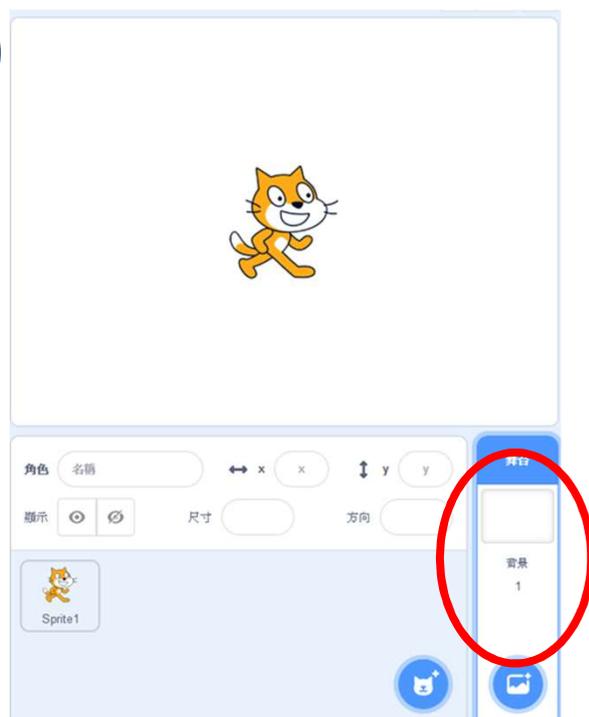
1. 請到 scratch.mit.edu 登入你的帳號。
2. 請按「創造」以建立一個新的專案。
3. 請把它命名為「DrawASquare」。
4. 你可以為你的專案更改背景。
5. 例如你可以選擇「Stars」。

1

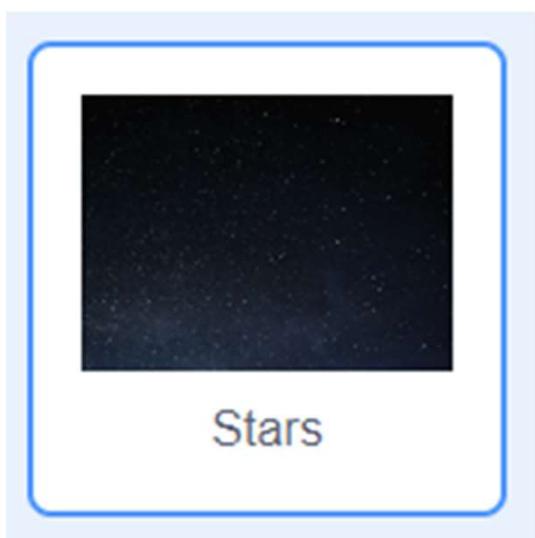
2

3

4



5



運用 Scratch 畫圖形

單元七
學習指引：第一教節

來編程：新增畫筆功能

你在開始繪畫前，需要新增 **畫筆** 元件到你的 Scratch 專案。

1. 請點擊左下角「**添加擴展**」圖示。



2. 請選擇「**畫筆**」擴充功能。



3. 請在「**程式**」標籤中的「**畫筆**」，查看可使用的指令方塊。



運用 Scratch 畫圖形

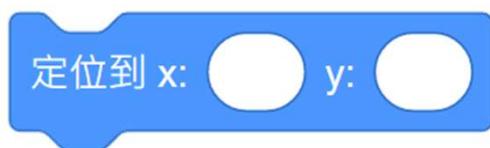
來編程：繪畫一條線

利用畫筆功能，只要移動角色，就能繪畫線條。

可參考附錄
P.28



1. 執行右方的指令方塊，你能畫出一條線嗎？
2. 再次點擊綠旗，這次有何不同？你又會怎樣解決呢？
提示：小貓的起點在哪？每次下筆應怎樣處理筆跡？



3. 改進程式：更改筆跡顏色為你最喜歡的顏色。
你亦可以變更畫筆的寬度。



測試及除錯

點擊綠旗，你能成功繪畫出一條線嗎？



運用 Scratch 畫圖形

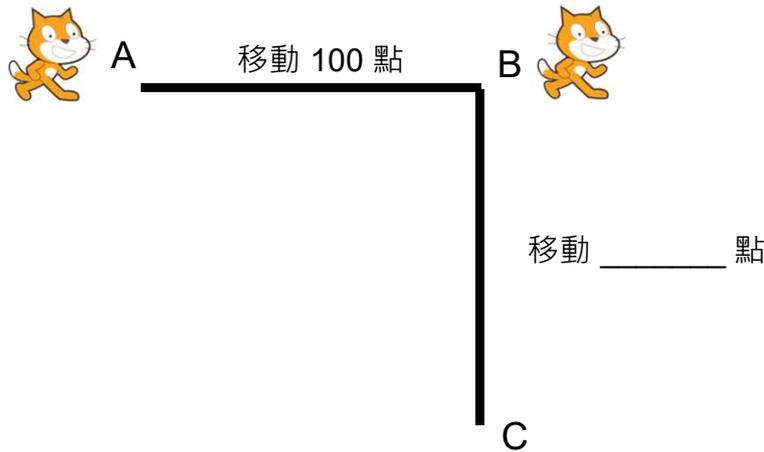
想一想、來編程：繪畫正方形

可參考附錄
P.29

繪畫了一條線後，我們又可怎樣繪畫一個正方形呢？

你在編程之前，可以完成下方尚未繪畫好的正方形嗎？

假設 Scratch 小貓從 A 移動 100 點到 B，那麼它從 B 移動了幾多點到 C 呢？



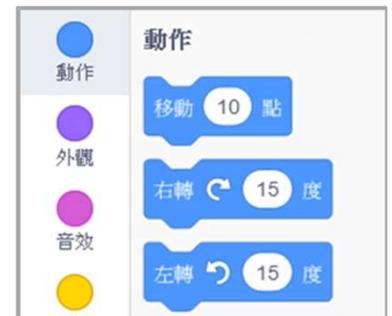
為了由 B 移動到 C，Scratch 小貓需轉動身體向下：



回到 Scratch 的編程環境，在「動作」欄，

拖出「右轉___度」指令方塊。

提示：嘗試 30 / 60 / 90 度，再測試以找到合適的方向。



找到正確的轉動角度後，繼續繪畫，將線延伸並畫出正方形吧！



測試及除錯

完成一小步編程後，你已經可以點擊綠旗，馬上測試，看看你的想法是否可行。然後繼續編程，直至繪畫出正方形。

繪畫一條線



繪畫正方形



運用 Scratch 畫圖形

單元七
學習指引：第一教節

想一想、來編程：使用重複指令方塊

你有沒有發現有些指令方塊重複出現？

這些指令方塊需要重複多少次才能畫出一個正方形呢？

可參考附錄
P.30



你能找到其他可替代方法畫出一個正方形嗎？例如 運用重複指令方塊的編程方式。
提示：從「控制」欄中拖出「重複」指令方塊。



測試及除錯

點擊綠旗進行測試！你能使用重複指令方塊繪畫一個正方形嗎？



運用 Scratch 畫圖形

單元七
學習指引：第二教節

你在這課堂將會學習如何使用多個正方形來繪畫一塊雪花圖形。

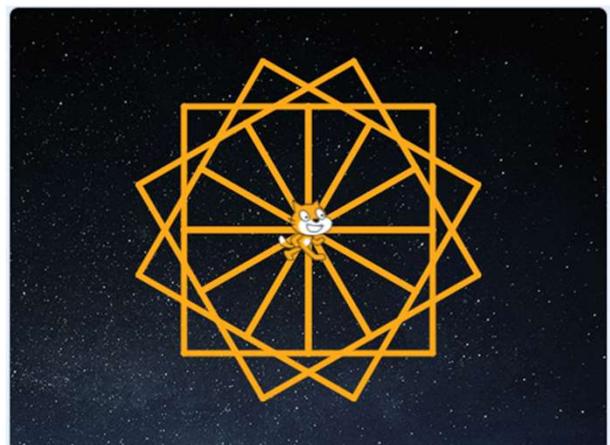
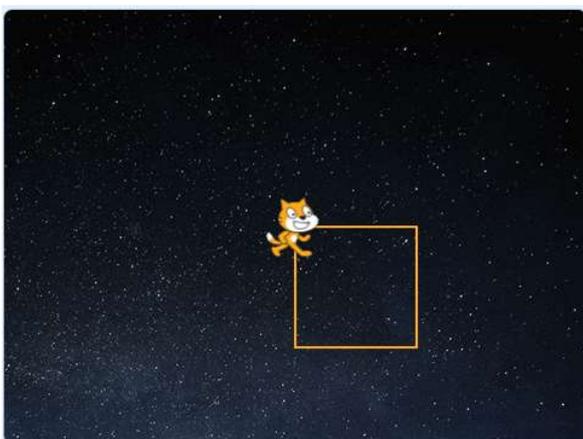


玩一玩

打開另一個 Scratch 的專案 <https://scratch.mit.edu/projects/737402437/>

點擊綠旗看看會發生什麼事情。

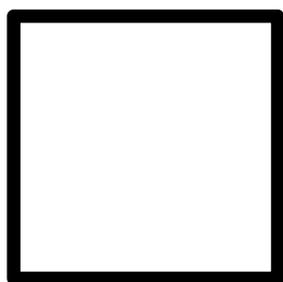
你知道正方形與雪花圖形的關係嗎？



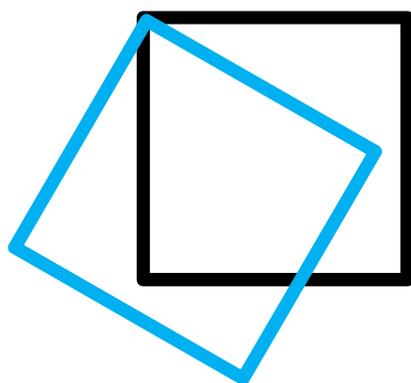
運用 Scratch 畫圖形

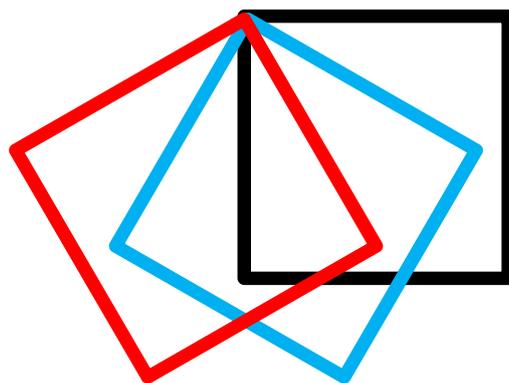
想一想：一塊雪花 = 多個正方形

讓我們拆解雪花，看看它是如何製作的。請在橫線上填寫該圖形含有多少個正方形：

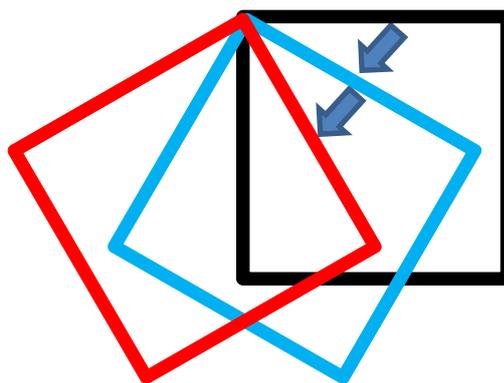
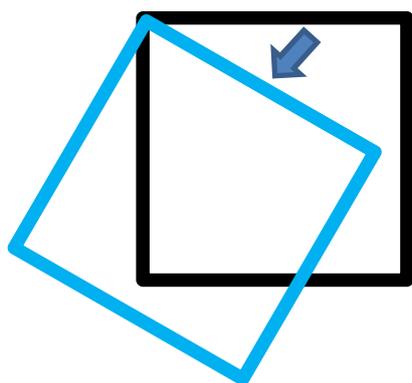


1





繪畫一個正方形，轉動一些角度，接着畫下一個，並不停重複。

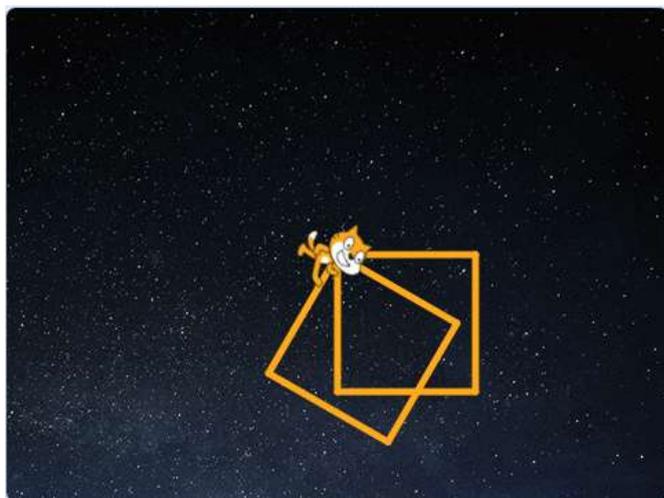
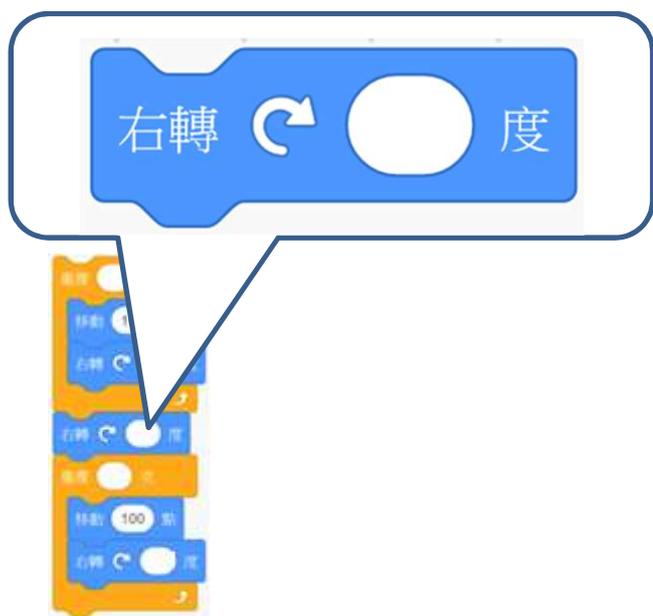


運用 Scratch 畫圖形

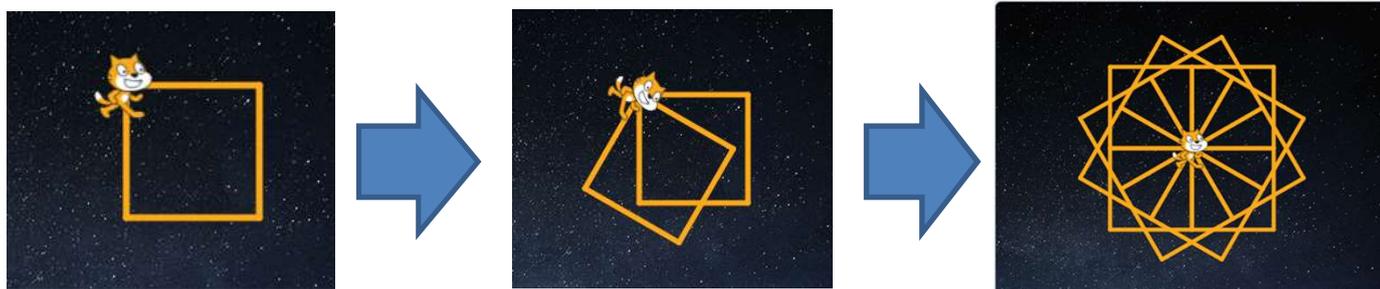
想一想、來編程：繪畫多個正方形

現在是時候在 Scratch 中嘗試了！把專案另存成複本，名為 **DrawASnowflake**（意思即繪畫一塊雪花）。

繪畫第一個正方形後，你認為它應該轉動多少度才可開始繪畫另一個正方形？
提示：可嘗試 30 度。



重複（如 12 次）剛才的步驟，直至你繪畫出一塊雪花。



測試及除錯

點擊綠旗進行測試！你能使用繪畫出多個正方形（一塊雪花）嗎？



運用 Scratch 畫圖形

想一想、來編程：繪畫多個正方形

你的指令方塊有多長？是否過於冗長呢？

可參考附錄 P.31

運用所學，我們可怎樣令它簡潔？

提示：嘗試「重複」指令方塊。

試想想，我們能否改變右轉角度和重複的次數，來繪畫不同的雪花呢？

```

重複 1 次
  移動 100 點
  右轉 90 度
重複 1 次
  右轉 90 度
  移動 100 點
  右轉 90 度
重複 1 次
  右轉 90 度
  移動 100 點
  右轉 90 度
重複 1 次
  右轉 90 度
  移動 100 點
  右轉 90 度
  
```



測試及除錯

再次測試！簡化後的指令方塊也能繪畫出相同圖案嗎？



運用 Scratch 畫圖形

想一想

你是否運用以下方法，以較簡潔的指令方塊，來繪畫多個的正方形呢？



這樣夠清楚嗎？



如果我們能直接叫電腦
「繪畫一個正方形」
(Draw a Square) 呢？



-
-
-

想一想

不插電活動：喝水

「喝水」這個動作很簡單，試寫出「喝水」所包含的詳細步驟：

A	喝水	
B	拿出水樽、 <hr/> <hr/>	

當你平時在課室想「喝水」時，你會怎樣詢問老師？嘗試以下兩種不同的發問方式：

- A. 老師，我可否「喝水」？
- B. 老師，我可否（所有「喝水」的詳細步驟）？

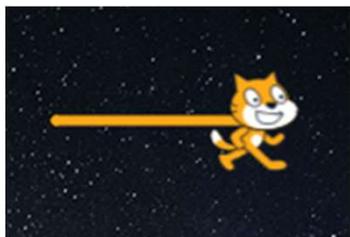
哪種方式較為簡潔？

同樣道理，我們應如何叫電腦「繪畫一個正方形」呢？回想一下你在 Scratch 裏運用了哪些指令方塊來繪畫正方形？詳細步驟是怎樣？

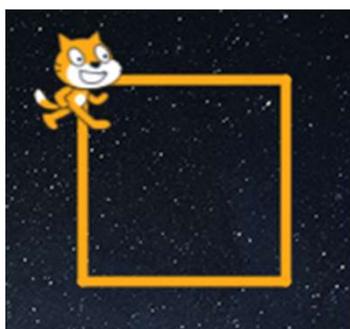
運用 Scratch 畫圖形

想一想

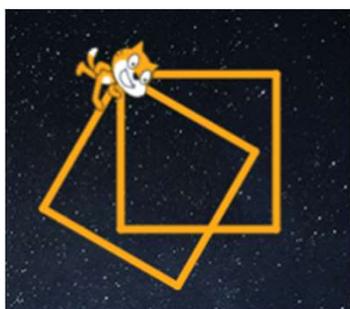
你能寫下我們繪畫一條線 / 正方形 / 多個正方形的步驟嗎？



繪畫一條線：
移動 _____ 點



繪畫正方形：
重複「繪畫一條線」的步驟 > 右轉
_____ 度 > 重複 _____ 次



繪畫多個正方形：
重複「繪畫 _____ 」的步驟 > 右轉
_____ 度 > 重複 _____ 次

運用 Scratch 畫圖形

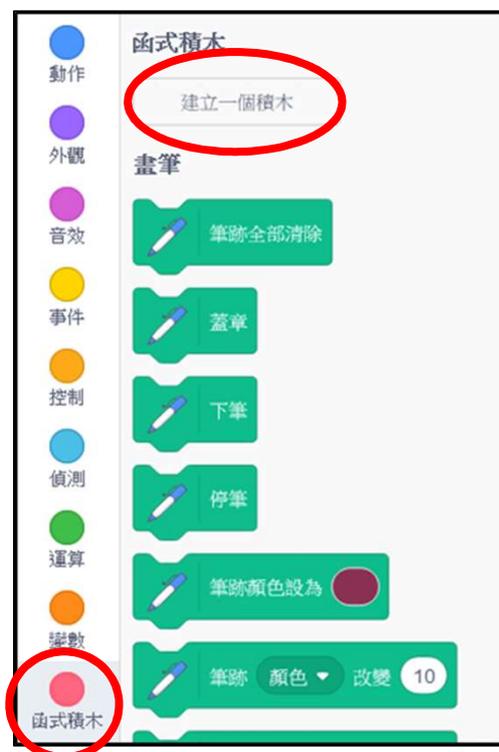
單元七
學習指引：第三教節

想一想、來編程：創建“DrawASquare”指令方塊

讓我們製作一個由我們自訂的指令方塊吧！

1. 請在「**函式積木**」欄點選「**建立一個積木**」。

1



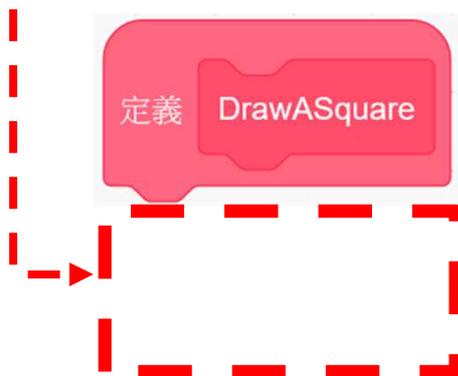
2. 輸入「**DrawASquare**」作為指令方塊的名稱，然後點擊「**確定**」。



想一想、來編程：創建 “DrawASquare” 指令方塊

創建了自訂指令方塊 **DrawASquare** 之後，你會看到「**定義 DrawASquare**」指令方塊。

想一想哪些指令方塊可以幫助你畫出正方形，把它們放置在 **DrawASquare** 指令方塊下，以此「**定義 DrawASquare**」。



知識建立：抽象化及模組化



抽象化指在特定的情景中，找出關鍵資訊並**暫不處理細節**的一種表達方式。

例如，自訂指令方塊（程序）（**procedure**）名為「畫一個正方形」（**Draw A Square**），這個**名字包含關鍵資訊**，表達了即將在這個模組裏所要做的事情。

模組化表示在電腦編程是指為該項**任務**編寫**詳細**的程式**指令**，讓它執行，並盡量保持這些程式指令可以**重用**。只需**編寫一次**，便能在程序中**到處使用**。例如，我們使用“**DrawASquare**”這個模組來繪畫多個正方形。

運用 Scratch 畫圖形

單元七
學習指引：第三教節

想一想、來編程：創建“DrawASquare”指令方塊

可參考附錄
P.32

創建自訂指令方塊不代表它會自行運行，所以我們需要呼叫它出來。

1. 你在「函式積木」欄裏，找到並拖出「DrawASquare」指令方塊。
2. 然後運用自訂指令方塊繪畫多個正方形。



測試及除錯

完成後便可馬上測試！你能運用自訂指令方塊繪畫雪花嗎？



運用 Scratch 畫圖形

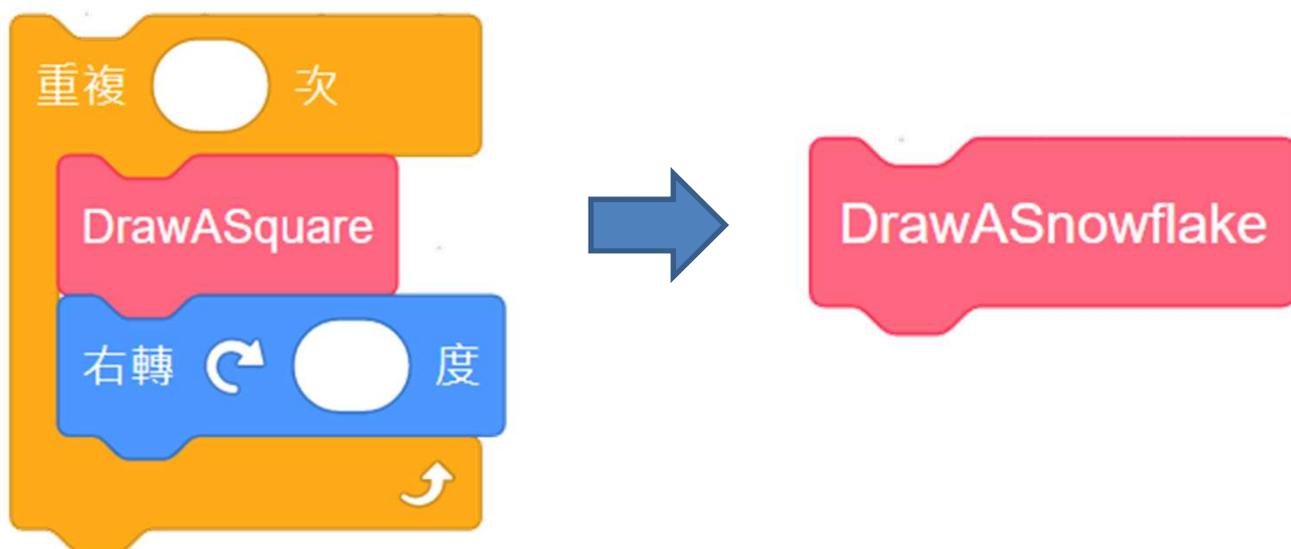
單元七
學習指引：第三教節

想一想、來編程：創建“DrawASnowflake”指令方塊

可參考附錄
P.33

最後，如果我們想要電腦直接「繪畫一塊雪花」呢？

檢查你的程式，創建另一個名為“DrawASnowflake”的自訂指令方塊來實現。



測試及除錯

點擊綠旗測試，看看你的雪花圖案是否仍然相同。



運用 Scratch 畫圖形

單元七
學習指引：第三教節

齊反思：「兩顆星星，一個希望」工作紙

專案名稱：_____ 創作者姓名：_____

請寫下在專案中你最喜歡的地方。



寫出一項你希望為專案加入或更改的東西，使專案變得更好。



重溫練習

1. 一名學生在他的專案中使用了以下指令方塊，當使用者點擊綠旗時會發生什麼事呢？



- A. 角色會繪畫出一塊雪花。
- B. 角色會繪畫出 4 個正方形。
- C. 角色會繪畫出 1 個正方形。
- D. 舞台被清除，角色會移到舞台中央。

運用 Scratch 畫圖形

重溫練習

2. 如果使用者按 3 下向右鍵會發生什麼事情呢？



- A. 角色向右移動 30 步。
- B. 角色移動到舞台中央。
- C. 角色在舞台上繪畫 3 個並排的正方形。
- D. 角色在舞台上繪畫 3 個相互重疊的正方形。

重溫主要元件

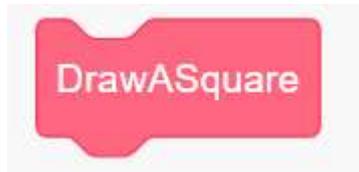
畫筆：讓角色（Scratch 小貓）變為一支筆，隨意在舞台上繪畫。

畫筆



重溫主要元件

函式指令方塊：以模組化和抽象順序建立電腦程式的指令方塊。



運用 Scratch 畫圖形

重溫主要概念和實踐

序列：電腦依照順序，逐一執行一連串步驟。



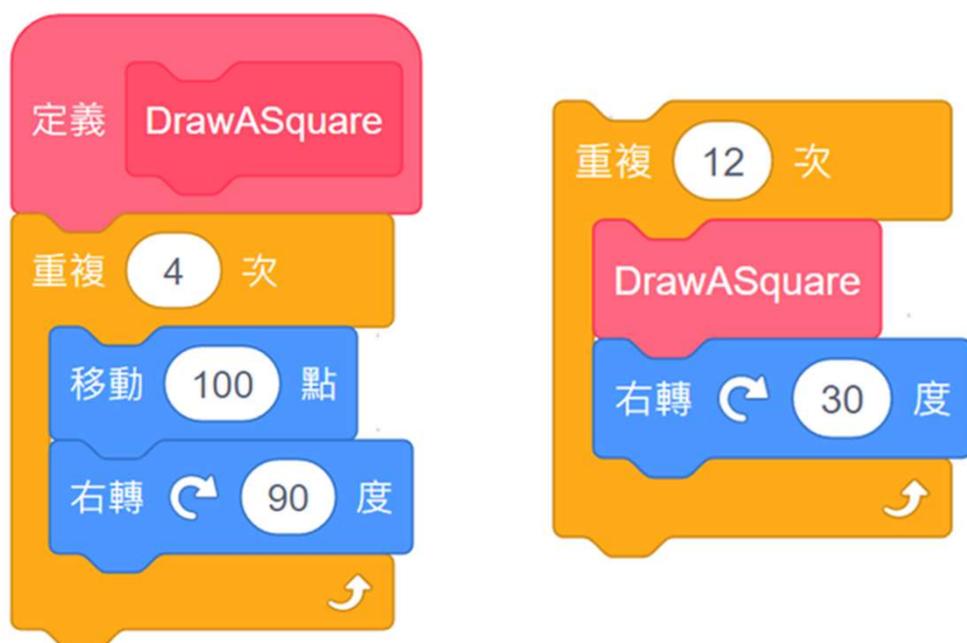
循環：循環是一個重覆的過程並藉此產生一系列的。我們可以在 Scratch 運用「重覆__次」引發重覆的動作。



重溫主要概念和實踐

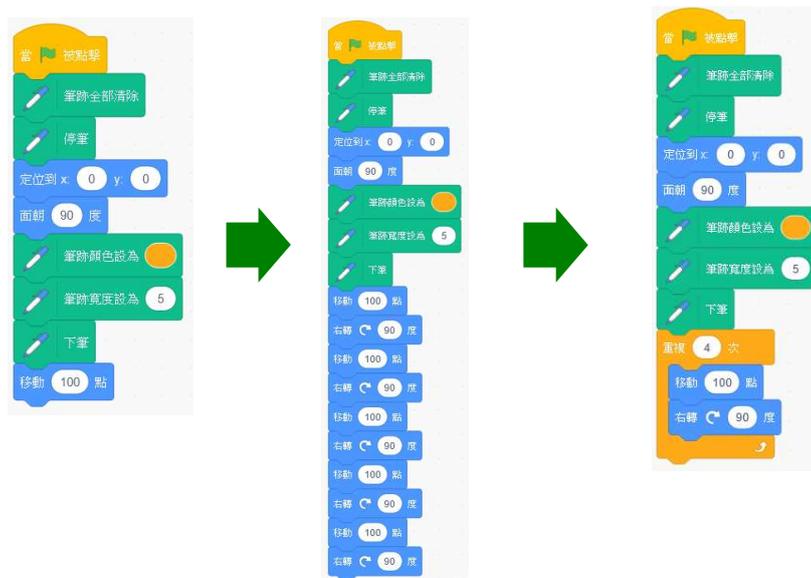
抽象化及模組化：「抽象化」在電腦編程是指在特定的情景中，找出關鍵資訊並暫不處理細節的一種表達方式。例如，我們在 Scratch 編程環境裏創建的一組自訂指令方塊（程序）名為 DrawASquare，這個名字包含關鍵資訊，表達了即將在這個模組裏所要做的事情。

「模組化」表示在電腦編程是指為該項任務（例如畫一個正方形編寫詳細的程式指令，讓它執行，我們通常在編寫詳細的程式指令過程，盡量保持這些程式指令可以重用。例如，我們使用“DrawASquare”這個模組來繪畫多個正方形。

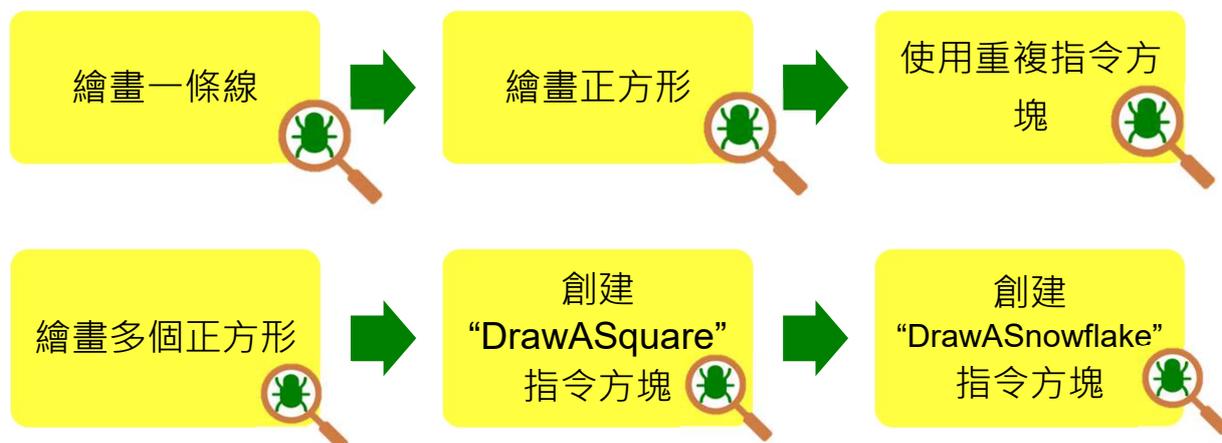


重溫主要概念和實踐

反覆構思及漸進編程：每次計劃一步編程工作，再計劃下一步。漸進意指每次編程都根據上一步建立了的基礎來改進，直至完成編程。



測試及除錯：測試電腦程序是一個檢查它能否按原本的設計進行運作的過程。除錯就是為程序找出錯誤的源頭並改正錯誤。



附錄

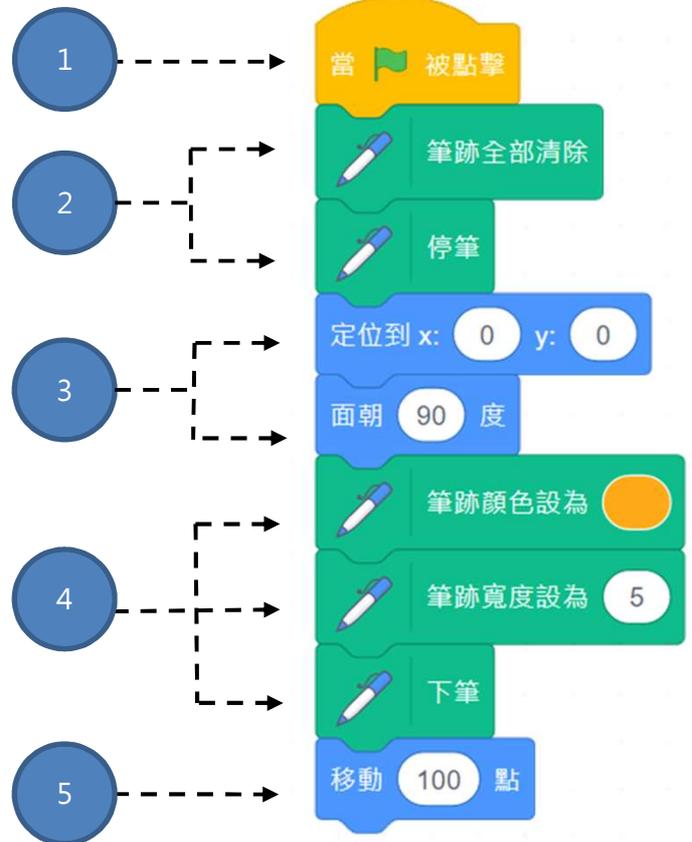
操作指南

運用 Scratch 畫圖形

來編程：繪畫一條線

見學習指引 P.5

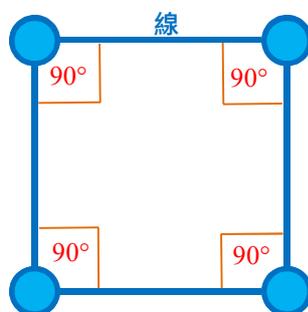
1. 從「事件」欄中拖出「當綠旗被點擊」的指令方塊。
2. 從「畫筆」欄中拖出「筆跡全部清除」及「停筆」，以清空畫面，並把它們加入到「當綠旗被點擊」的指令方塊下。
3. 為了讓角色返回起始位置，請從「動作」欄中拖出「面朝 90 度」及「定位到x:0 y:0」的指令方塊，並把它們插入剛才的指令方塊中。
4. 從「畫筆」欄中拖出「筆跡顏色設為」、「筆跡寬度設為」及「下筆」的指令方塊，並把它們放在剛才的指令方塊之下，以讓角色準備繪畫。
5. 從「動作」欄新增一個「移動 100 點」的指令方塊，來指示角色畫線。



運用 Scratch 畫圖形

想一想、來編程：繪畫正方形

通過直角（90度）連接四條線以繪畫一個正方形。



見學習指引 P.6

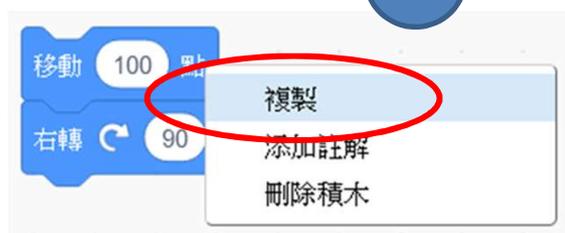
繪畫一個正方形，需要在繪畫下一條線前先把角度轉 90 度。

1. 從動作欄中拖出右轉 90 度的指令方塊，並把它放置於移動 100 的指令方塊下。

改為 90 度



2. 用滑鼠右鍵點擊移動 100 點的指令方塊，以複製它和右轉 90 度的指令方塊，然後把所複製的指令方塊移動到步驟一的指令方塊下。



3. 重覆步驟二兩次。

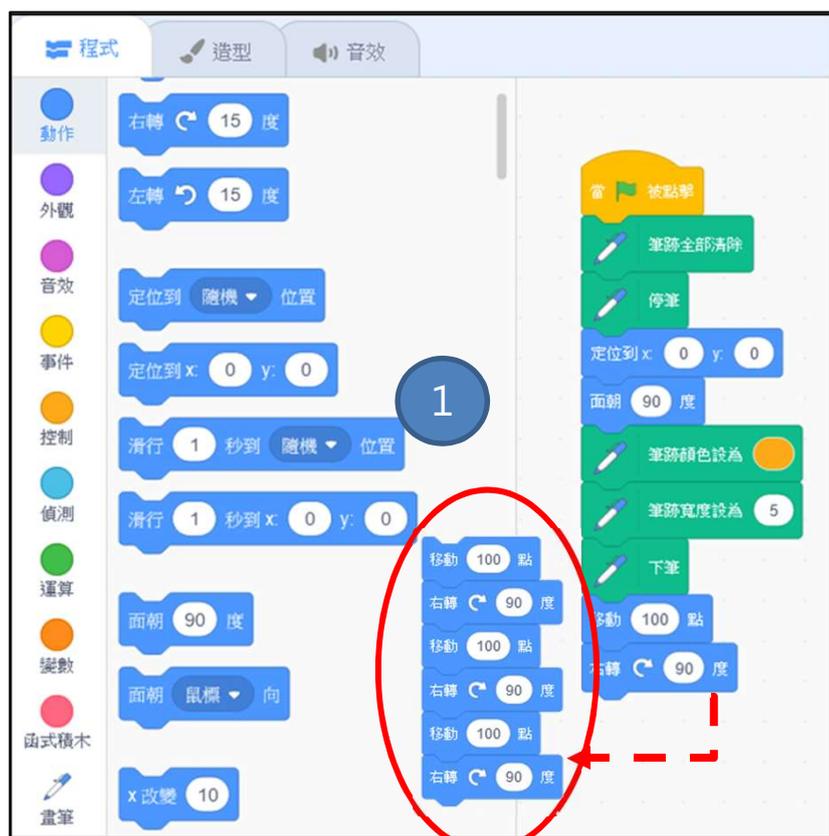


運用 Scratch 畫圖形

想一想、來編程：使用重複指令方塊

見學習指引 P.7

1. 移除重複的三組指令方塊：移動 100 點及右轉 90 度。



2. 從控制欄中拖出一個重複指令方塊，並把重複的次數改為「4」。
3. 把移動 100 點及右轉 90 度的指令方塊插入重複指令方塊中。

運用 Scratch 畫圖形

想一想、來編程：繪畫多個正方形

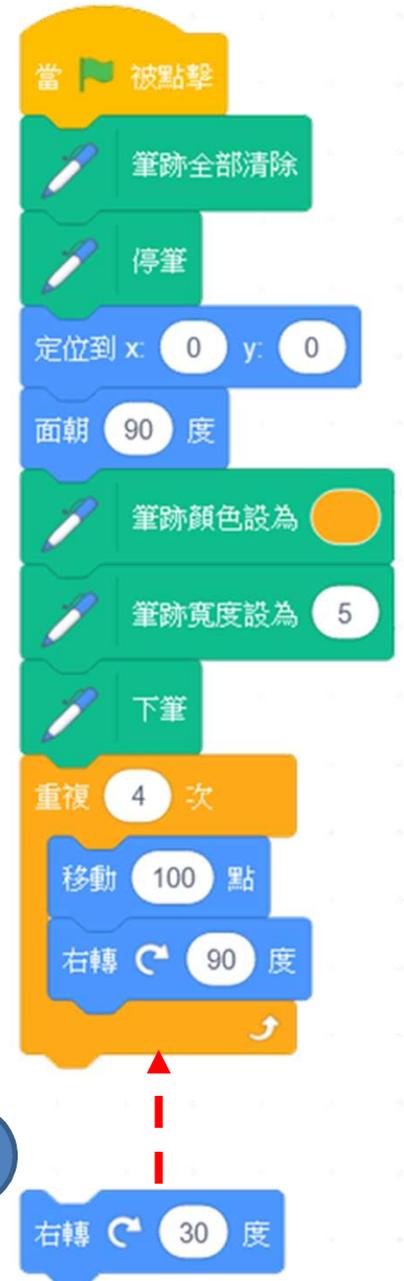
見學習指引
P.11

使用多個正方形來繪畫雪花，你需要：

1. 從**動作**欄中拖出**右轉 15 度**的指令方塊，然後把它放在**重複**指令方塊下，並把度數更改為**30 度**。
2. 複製**重複**的指令方塊及**右轉 30 度**的指令方塊。



3. 把複製的指令方塊放到**右轉 30 度**之下。



重複剛才的步驟，直至你繪畫出一塊雪花。

運用 Scratch 畫圖形

想一想、來編程：創建“DrawASquare”指令方塊

拖動可以幫助你畫出正方形的指令方塊，把它們放置在 DrawASquare 指令方塊下。

見學習指引
P.17



1. 在你繪畫一個新的正方形之前，角色需要旋轉 30 度。所以，把**右轉 30 度**加入到 **DrawASquare** 的指令方塊中。
2. 你需要畫出 12 個正方形來組成一個雪花，所以，從控制欄中拖出**重複**指令方塊，並把 **DrawASquare** 及**右轉 30 度**指令方塊插入其中。記得要把數字更改為 **12** 來繪畫 12 個正方形。
3. 最後，把**重複**指令方塊拖動至點擊綠旗下方。



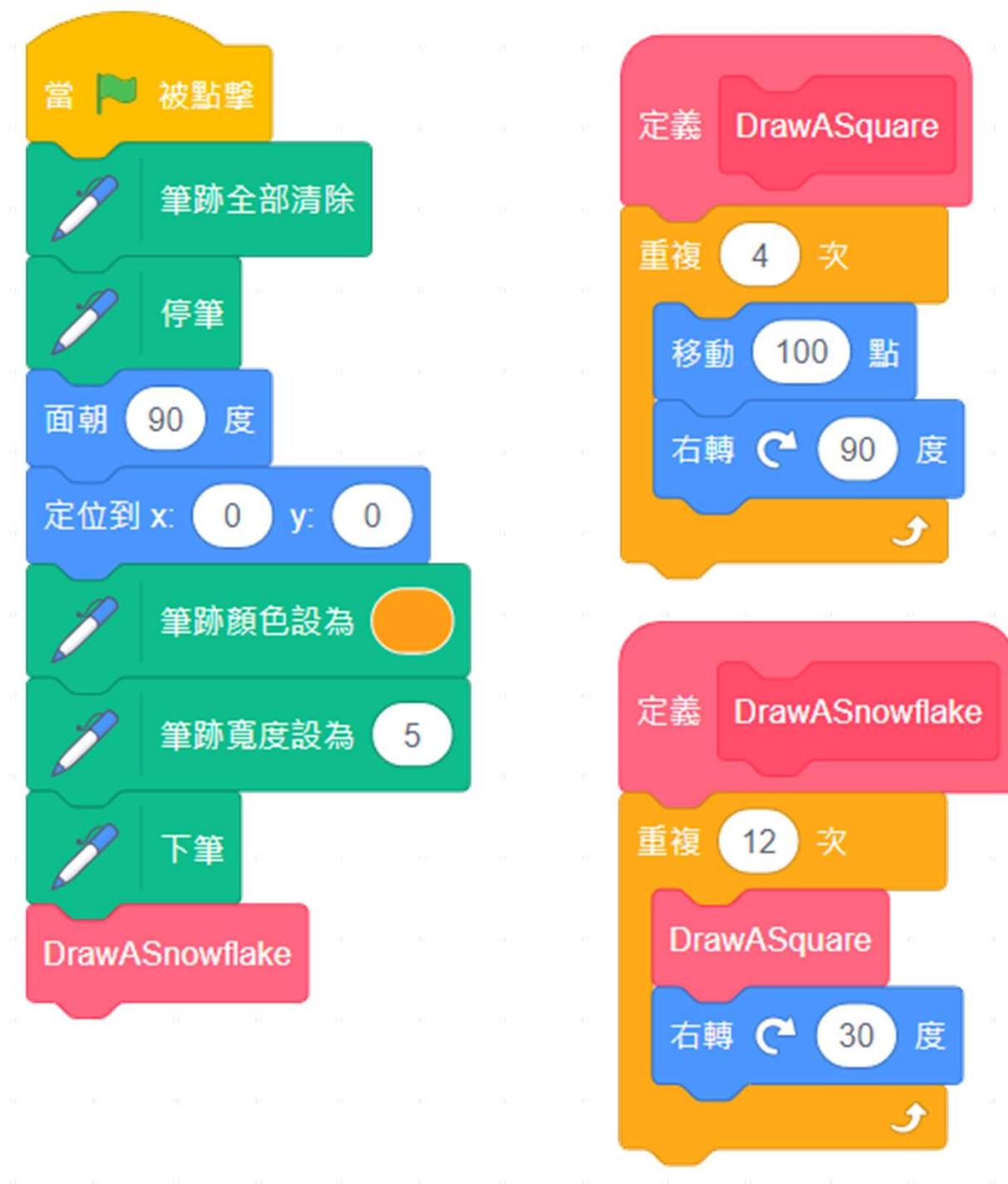
運用 Scratch 畫圖形

單元七
學習指引：第三教節

想一想、來編程：創建“DrawASnowflake”指令方塊

創建並運用“DrawASnowflake”指令方塊使程式更為簡潔。

見學習指引
P.18



單元八：運用 Scratch 設計圖案 (延伸單元) 學習指引

目錄

第一教節

玩一玩 S8-1

想一想

繪畫正方形 S8-3

繪畫雪花 S8-4

繪畫多個雪花 S8-5

來編程

繪畫多個雪花 S8-6

想一想、來編程

運用 DrawASnowflake 指令方塊 S8-7

下筆及停筆 S8-8

第二教節

想一想、來編程

繪畫大小不同的雪花 S8-9

在自訂指令方塊中添加輸入 (數字或文字) 方塊 S8-10

齊反思 S8-13

重溫練習 S8-14

重溫主要元件 S8-16

重溫主要概念和實踐

S8-17

附錄 - 操作指南

S8-20

運用 Scratch 設計圖案

單元八
學習指引：第一教節

你在本單元將學習如何在舞台上的不同位置製作更多的雪花。此外，你將在你的專案中改變雪花的大小。



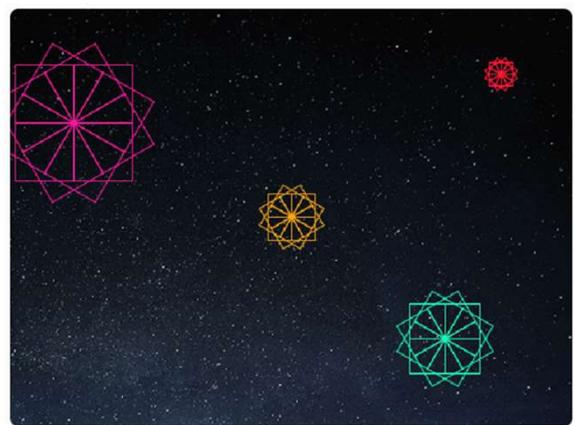
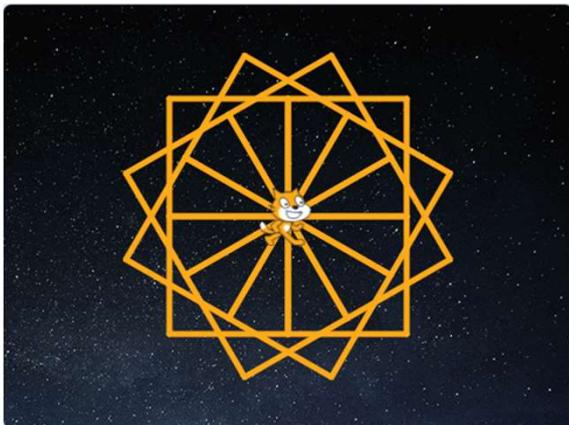
玩一玩

請在 <https://scratch.mit.edu/projects/737985288/> 打開這 Scratch 的專案。

點擊綠旗看看會發生什麼事情。

再次點擊綠旗，看看那些雪花是否在相同位置。

這一次你能看到 Scratch 小貓嗎？

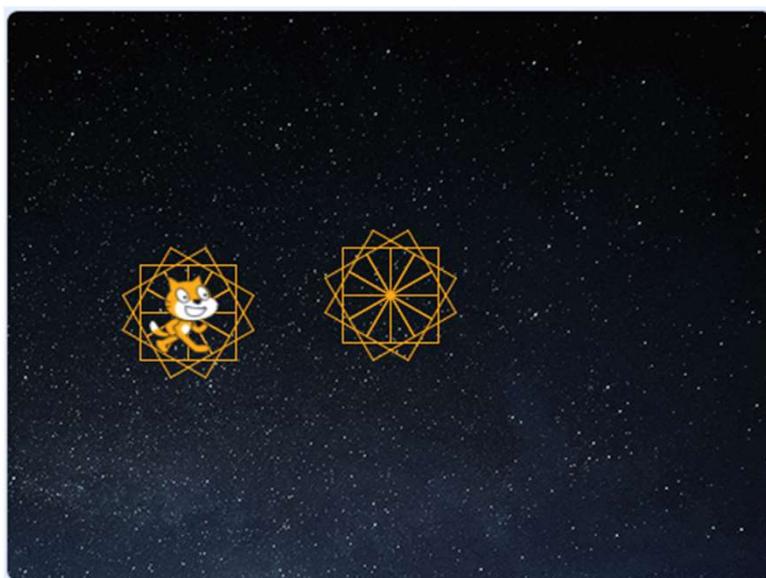


運用 Scratch 設計圖案

單元八
學習指引：第一教節

想一想

對於在舞台上的不同位置製作多個雪花，你有任何想法嗎？



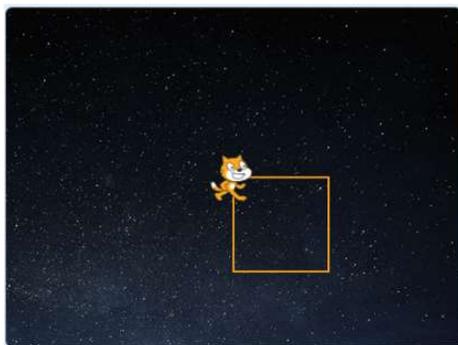
1. 請在 scratch.mit.edu 登入你的帳號。點選右上方你的名字，然後進入「我的東西」，並開啟你單元七「DrawASnowflake」的專案。
2. 從「檔案」中選擇「另存成複本」，然後將名稱更改為「DrawMultipleSnowflakes」，並儲存你的專案。



運用 Scratch 設計圖案

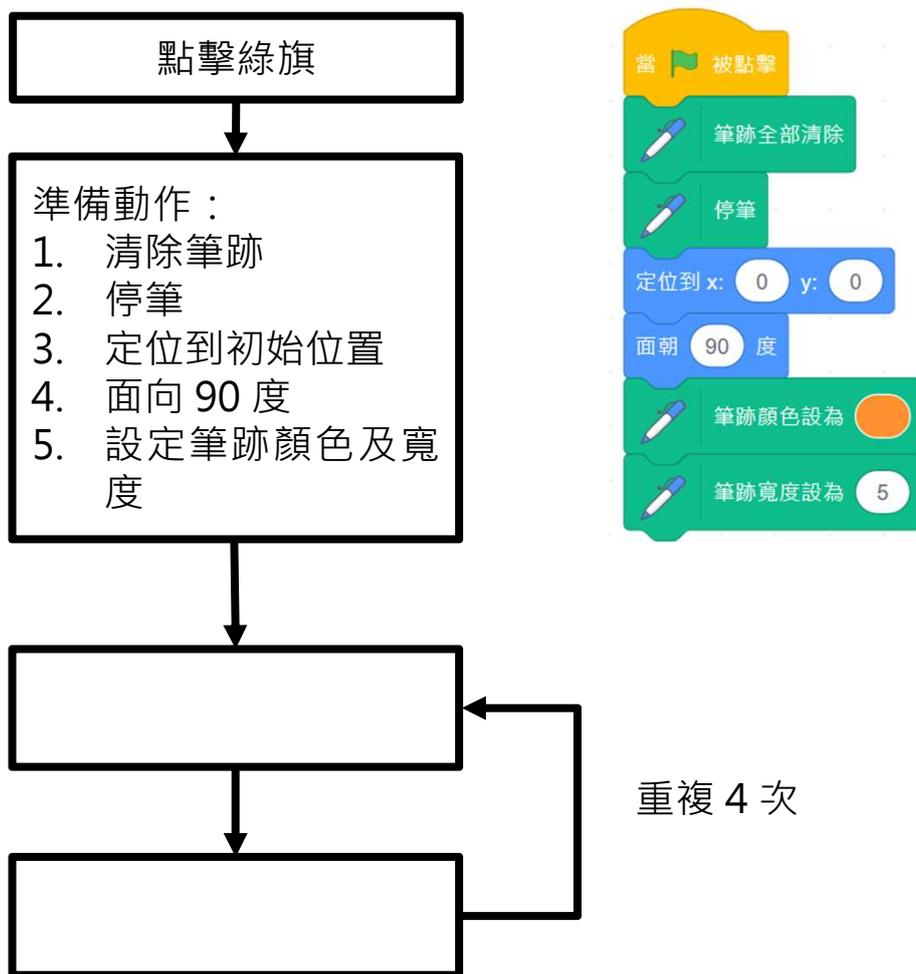
想一想：繪畫正方形

回顧上個單元學到的知識，你還記得怎樣繪畫正方形嗎？



A. 右轉 90 度

B. 繪畫直線



運用 Scratch 設計圖案

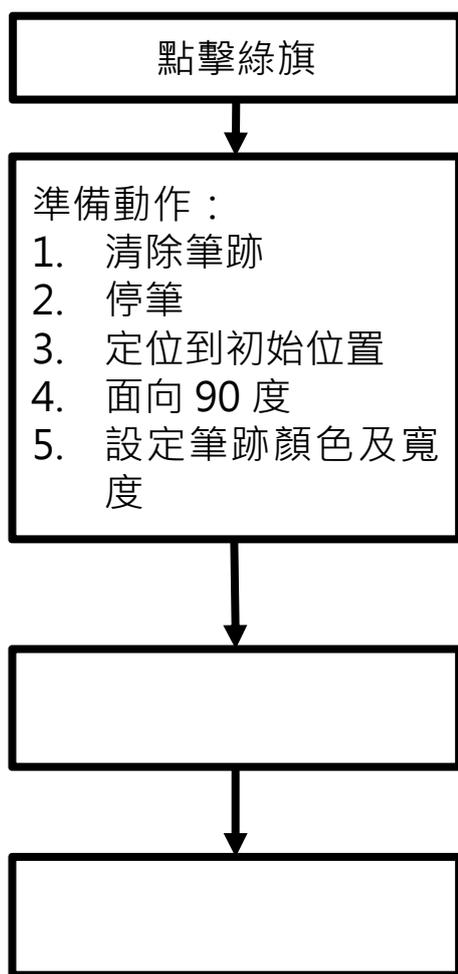
想一想：繪畫雪花

試把繪畫正方形引伸到繪畫雪花。



A. 右轉 30 度

B. 繪畫正方形



重複 12 次

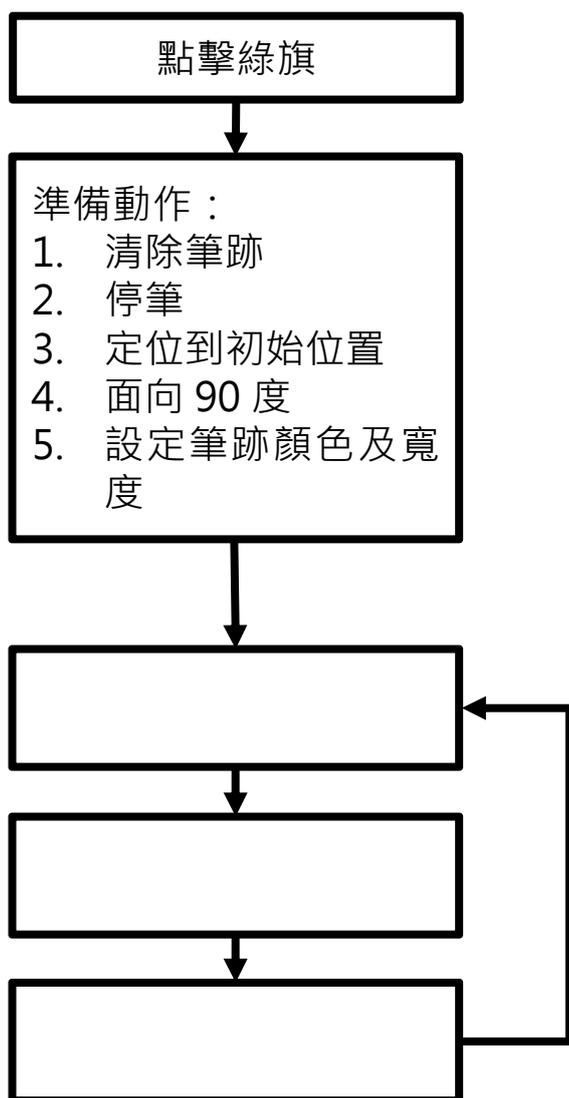
運用 Scratch 設計圖案

想一想：繪畫多個雪花

怎樣繪畫多個雪花呢？



A. 定位到隨機位置	B. 繪畫雪花	C. 改變畫筆顏色
------------	---------	-----------



重複 4 次

運用 Scratch 設計圖案

來編程：繪畫多個雪花

完成繪畫一片雪花後，角色應**移動**到一個**新的位置**。齊來編程吧！

提示：查看「動作」欄。

可參考附錄
P.21



當角色移動到新的位置時，你會如何繪畫第二片雪花呢？

提示：重複使用方塊以添加更多雪花。



點擊綠旗，看看雪花是否畫在不同的位置。
然後看看你的指令方塊，你能讓它更為簡潔嗎？



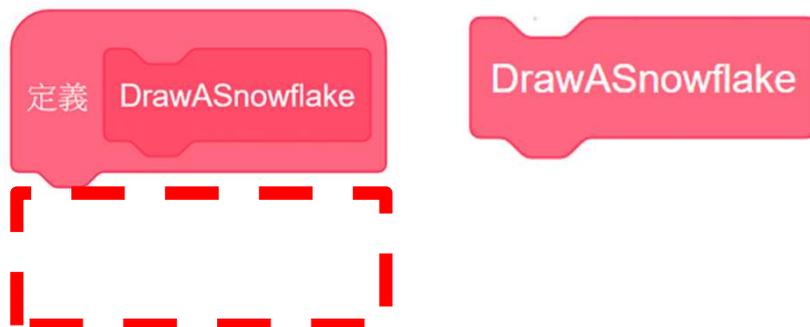
繪畫多個雪花



想一想、來編程：運用 DrawASnowflake 指令方塊

可參考附錄
P.22

你應該使用了一些重複的步驟來繪畫圖案。回顧上一單元學到的 DrawASnowflake 指令方塊。



這次你能運用「DrawASnowflake」指令方塊來簡化編程並繪畫多個雪花嗎？如果要在舞台上的不同位置繪畫多個雪花，你需要將角色移動到不同的位置。

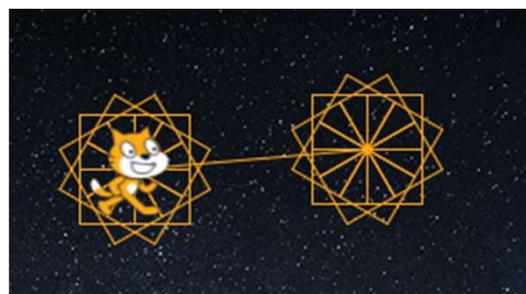
你想移動它到哪裏？是特定的 X / Y 座標位置，還是隨機位置？

提示：



測試及除錯

簡化後的指令方塊能做到相同效果嗎？另外，你有沒有看到兩片雪花之間有一條線呢？為什麼會發生這件事？



運用 Scratch 設計圖案

單元八
學習指引：第一教節

想一想、來編程：下筆及停筆

為什麼兩片雪花之間有一條線？

可參考附錄
P.23

畫筆是否保持在「下筆」？

繪畫一個正方形時，保持「下筆」，繪畫後便「停筆」。

查看「畫筆」欄，當中有哪一些指令方塊能協助你？
然後，請你更新「DrawASqaure」指令方塊。

提示：



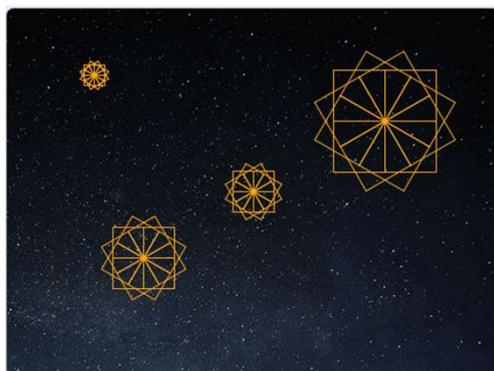
測試及除錯

點擊綠旗並進行測試吧！畫完一個雪花後，那條線消失了嗎？



想一想、來編程：繪畫大小不同的雪花

想一想，我們如何在不同的地方做出不同大小的雪花？



為了製作各種大小的雪花，我們需要改變組成雪花的正方形大小。

1. 在「定義 DrawASquare」指令方塊中，由 50 點更改為「移動（例如 30）點」。



2. 點擊綠旗，看看畫了什麼。

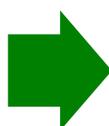
3. 如你覺得圖案太小無法看見，你可以選擇隱藏角色。



測試及除錯

如你以改變「移動（ ）點」來改變正方形的大小，所有雪花的大小都會被影響。

繪畫不同大小的雪花



想一想、來編程：在自訂指令方塊中添加輸入（數字或文字）方塊

怎麼才能畫出每一片大小都不同的雪花？你有什麼好主意嗎？

你可以為自訂指令方塊輸入參數，以製作各種大小的正方形和雪花。

要為「DrawASquare」指令方塊輸入參數，你需要：

1. 右鍵點擊「函式積木」欄中的「DrawASquare」指令方塊，並點擊編輯。
2. 然後點擊「添加輸入方塊」。

1



3. 輸入「size」以命名所輸入的參數，然後點擊「確定」按鈕儲存「DrawASquare」指令方塊。

3



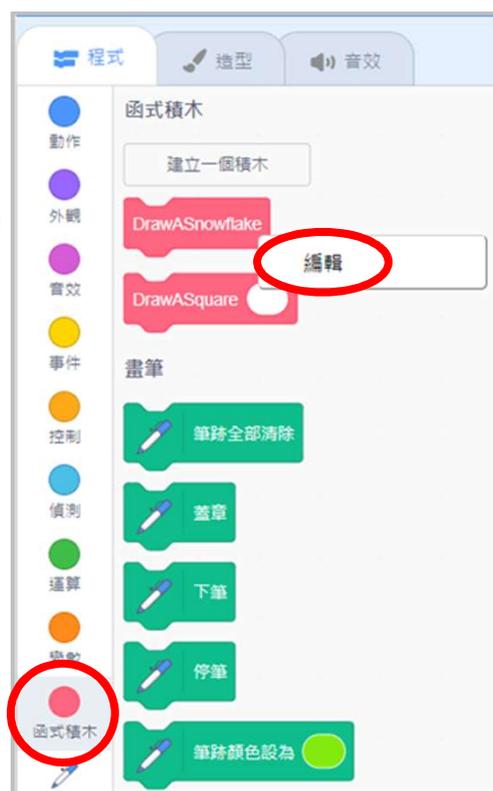
運用 Scratch 設計圖案

想一想、來編程：在自訂指令方塊中添加輸入（數字或文字）方塊

同樣地，讓我們也對「DrawASnowflake」指令方塊添加輸入參數吧！

1. 右鍵點擊「函式積木」欄中的「DrawASnowflake」指令方塊，並點擊編輯。
2. 然後點擊「添加輸入方塊」。

1



2



3. 輸入「size」以命名所輸入的參數，然後點擊「確定」按鈕儲存「DrawASnowflake」指令方塊。

3

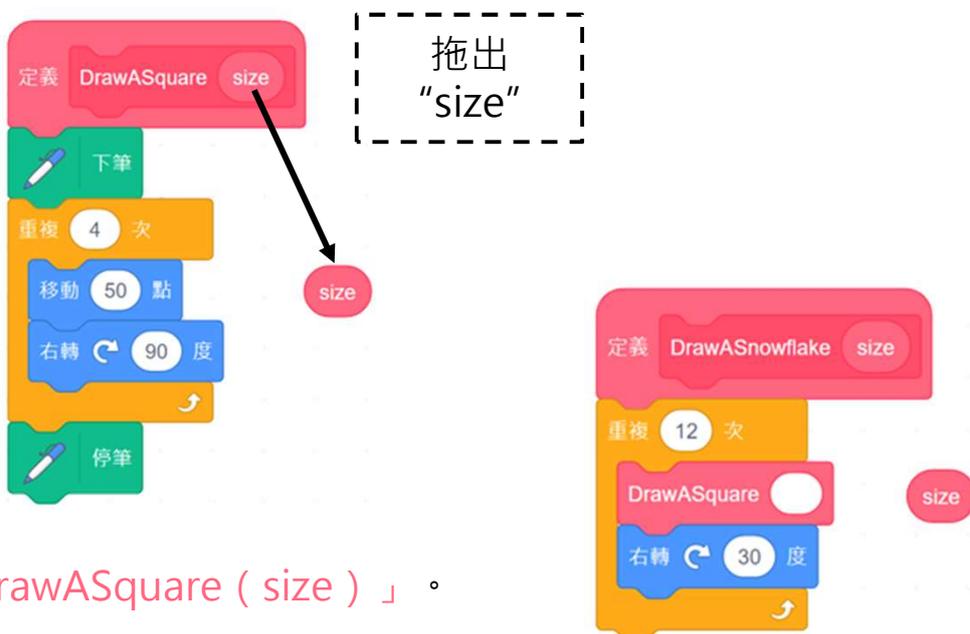


想一想、來編程：在自訂指令方塊中添加輸入（數字或文字）方塊

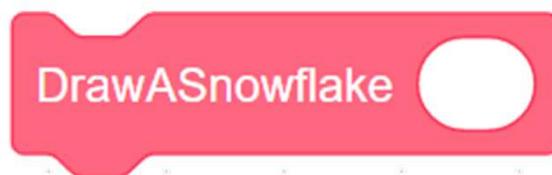
設置「DrawASquare」和「DrawASnowflake」指令方塊的大小。

可參考附錄
P.24

1. 從「DrawASquare (size)」中拖出「size」。我們應該把「size」指令方塊放在哪裏？哪一個指令方塊該被替換？



2. 你亦應更新「DrawASquare (size)」。
3. 繪畫不同大小的雪花時，請在「DrawASquare ()」指令方塊中，輸入不同的數值。

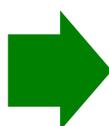


測試及除錯

點擊綠旗，看看畫了什麼。

現在，你可以在你的專案中再添加三個不同大小和位置不同的雪花嗎？

繪畫不同大小的雪花



在自訂指令方塊中添加輸入（數字或文字）方塊

運用 Scratch 設計圖案

單元八
學習指引：第二教節

齊反思：「兩顆星星，一個希望」工作紙

專案名稱：_____ 創作者姓名：_____

請寫下在專案中最喜歡的地方。

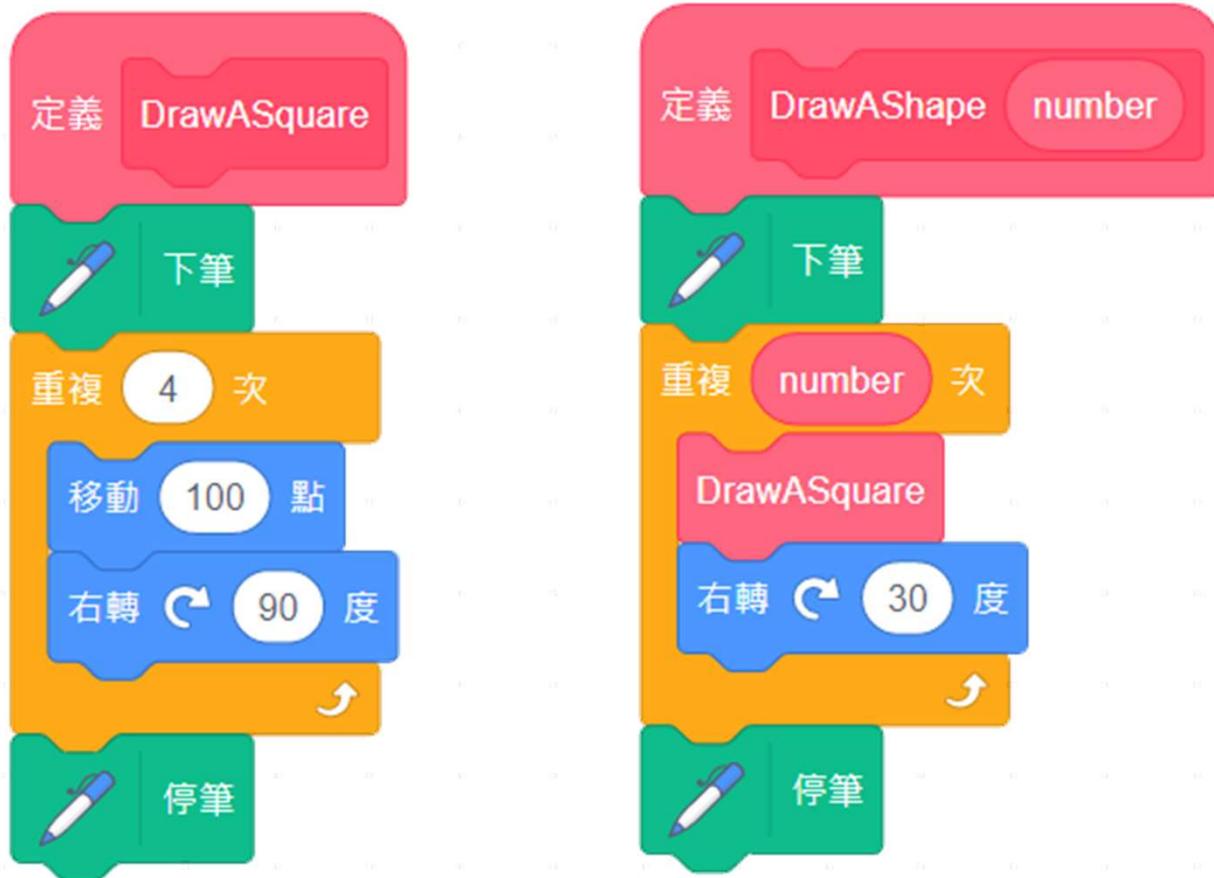


寫出一項你希望為專案加入或更改的東西，使專案變得更好。

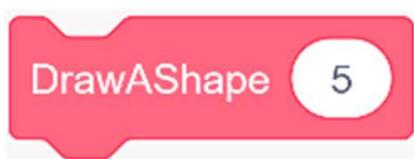


重溫練習

1. 一名學生正使用以下的自訂指令方塊製作專案。



如該名學生使用

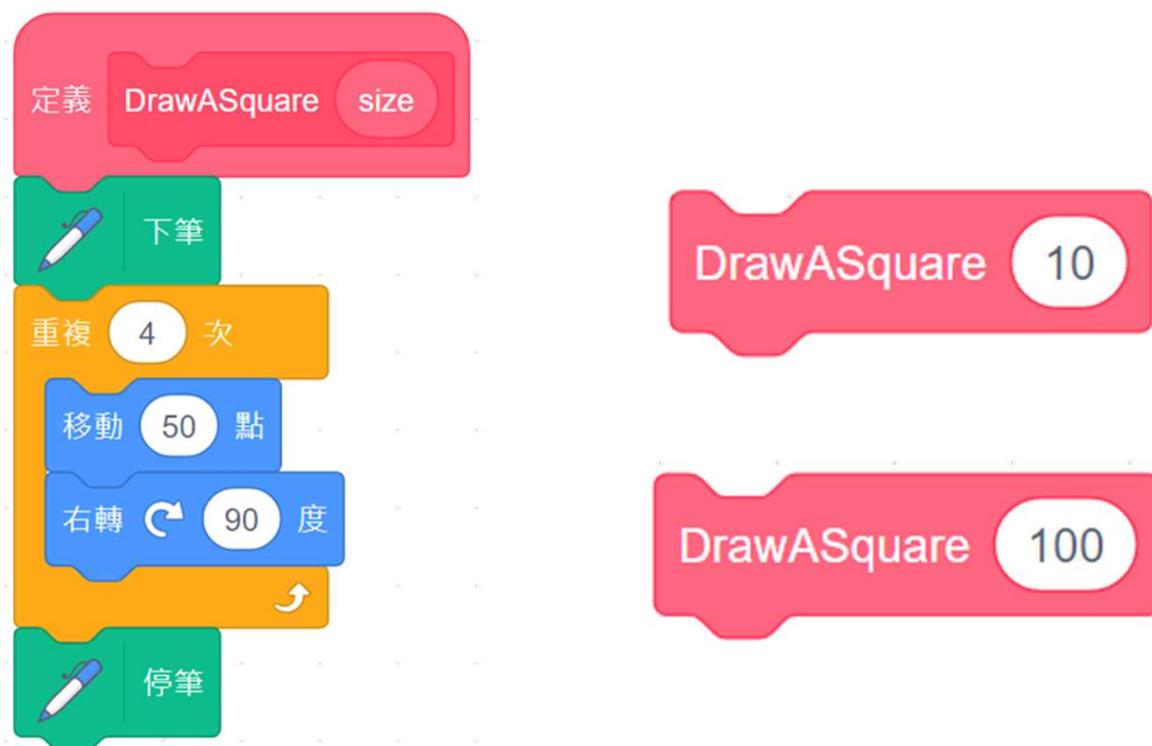


會發生什麼事？

- A. 繪畫了一個大小為 5 的雪花。
- B. 畫了 5 片雪花。
- C. 繪畫一個包含 5 個旋轉正方形的圖案。
- D. 什麼也沒有畫，因為筆被提起了。

重溫練習

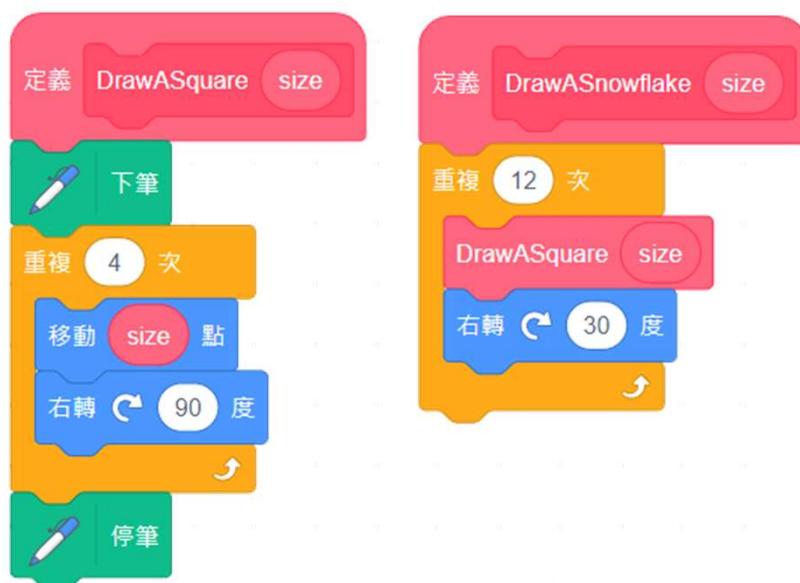
2. 一名學生想繪畫不同大小的圖案。但是，無論他使用 DrawASquare (10)，還是 DrawASquare (100)，畫出來的正方形都是相同。你能協助他除錯嗎？



- A. 重新命名「DrawASqaure」指令方塊為「DrawASnowflake」指令方塊。
- B. 刪除「DrawASqaure」指令方塊中的「size 參數」。
- C. 「重複 size 次」而非「重複 4 次」。
- D. 用「size 參數」替換「移動 50 點」指令方塊中的「50」。

重溫主要元件

附帶參數的函式積木：透過輸入參數 “size”（大小），我們可繪畫大小不同的正方形及雪花圖案。



重溫主要概念和實踐

序列：這是指執行編程指令的次序。錯誤的次序會使程式無法正確執行。



循環：循環是一個重覆的過程並藉此產生一系列的輸出。我們可以在 Scratch 運用「重複__次」引發重覆的動作。



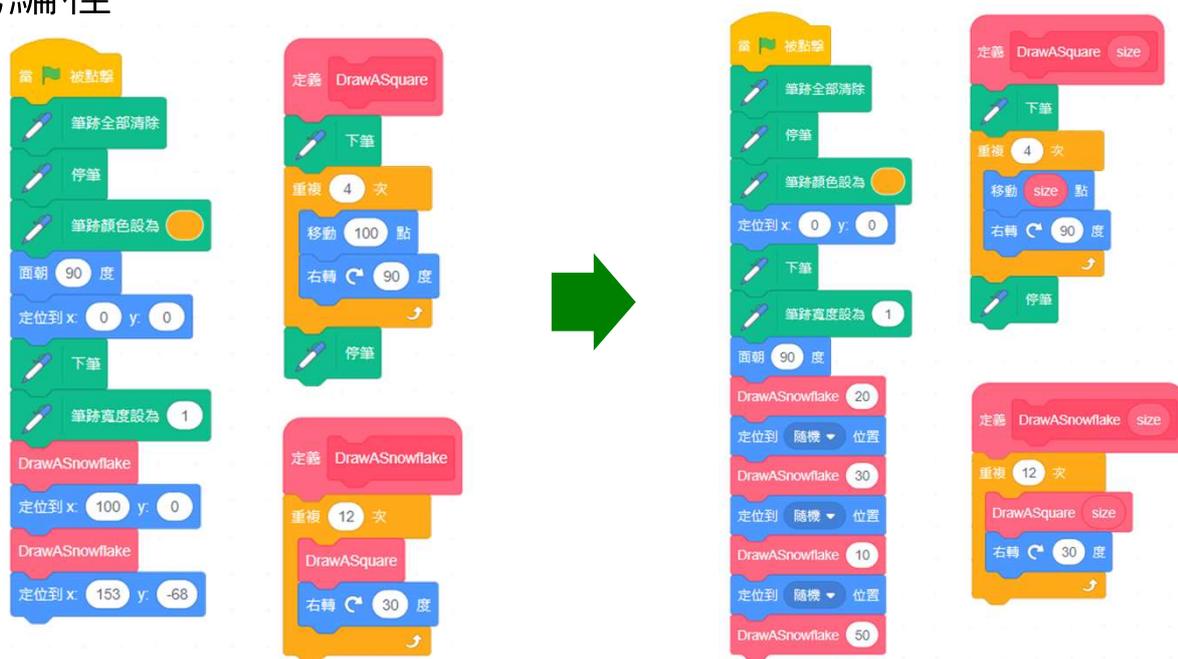
重溫主要概念和實踐

抽象化及模組化：「抽象化」在電腦編程是指在特定的情景中，找出關鍵資訊並暫不處理細節的一種表達方式。例如，我們在 Scratch 編程環境裏創建的一組自訂指令方塊（程序名為 DrawASnowflake，這個名字包含關鍵資訊，表達了即將在這個模組裏所要做的事情。「模組化」表示在電腦編程是指為該項任務（例如畫一個正方形）編寫詳細的程式指令讓它執行，我們通常在編寫詳細的程式指令過程，盡量保持這些程式指令可以重用。例如，我們使用“DrawASnowflake”這個模組來繪畫多個正方形。此外，我們在自訂指令方塊加入一個變量名為“size”，即「參數」。讓我們能稍為改動模組。運用參數“size”，我們可以創建不同大小的正方形和雪花圖案。



重溫主要概念和實踐

反覆構思及漸進編程：每次計劃一步編程工作，再計劃下一步。每次編程都根據上一步建立了的基礎來改進，直至完成編程。



測試及除錯：測試電腦程序是一個檢查它能否按原本的設計進行運作的過程。除錯就是為程序找出錯誤的源頭並改正錯誤。



附錄

操作指南

運用 Scratch 設計圖案

來編程：繪畫多個雪花

見學習指引 P.6

1. 要將角色移動到新的位置，你需要從「動作」欄中拖動出「定位到 x:0 y:0」的指令方塊，再把它拖到所有的指令方塊之下，並將「x」更改為「100」。

1



2. 複製繪畫雪花時所使用的重複指令方塊。

2



3. 拖出複製的「重複」指令方塊到定位到「x:100 y:0」的指令方塊下。

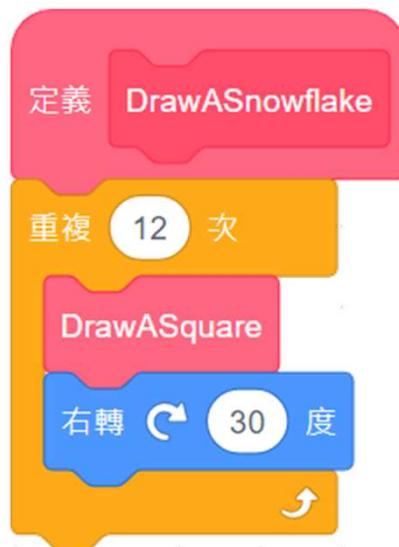
3



想一想、來編程：運用 DrawASnowflake 指令方塊

見學習指引 P.7

重溫：拖動繪畫雪花指令方塊並將它們插入到**定義 DrawASnowflake**指令方塊下。



要在舞台上的不同位置繪畫多個雪花，你需要將角色移動到不同的位置。

1. 從**動作欄**定位到 **x: y:** 的指令方塊，並把它放在現有指令方塊之下。
2. 在**定位到 x: y:** 的指令方塊中改變 **x** 和 **y** 的值，以改變雪花出現的位置。
3. 拖動指令方塊 **DrawASnowflake** 至**定位到 x: y:** 的指令方塊下。



如果想添加更多雪花，你只需重複步驟 1 至 3。

你也可嘗試使用**定位到隨機位置**以替代繪畫在特定位置。



運用 Scratch 設計圖案

單元八
學習指引：第一教節

想一想、來編程：下筆及停筆

見學習指引 P.8

當運行「DrawASquare」程序時，繪畫一個正方形時，保持「下筆」，繪畫後便「停筆」。

如果你認為這個圖案太大或太少，你可以改變所移動的步數（如：50）。

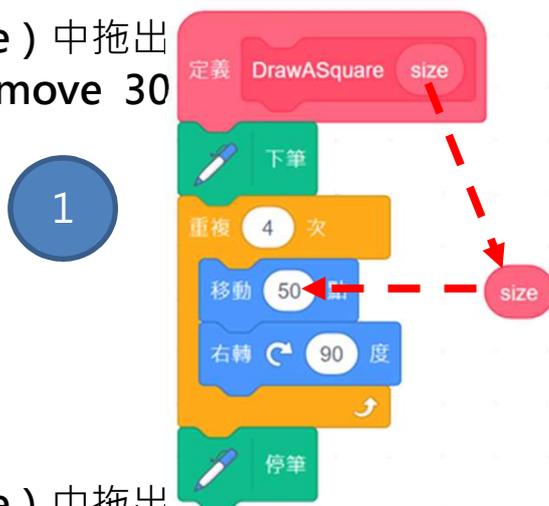


想一想、來編程：在自訂指令方塊中添加輸入（數字或文字）方塊

見學習指引
P.12

設置 DrawASquare 和 DrawASnowflake 指令方塊的大小。

1. 從自訂指令方塊 DrawASnowflake (size) 中拖出 size，並以此替代指令方塊「移動30點 move 30 steps」中的 30。



2. 從自訂指令方塊 DrawASnowflake (size) 中拖出 size，並把它放進 DrawASnowflake (size) 的空格內。



3. 為 DrawASnowflake () 指令方塊的每個空格填入三個不同大小的值。



Scratch 專題習作 學習指引

目錄

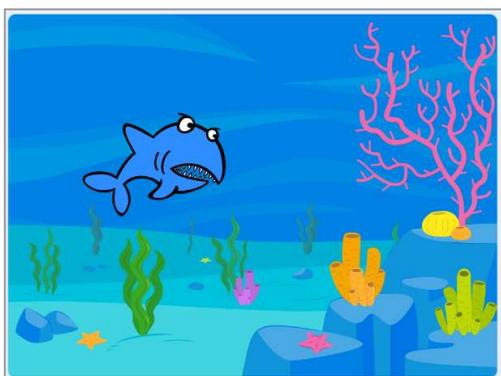
想一想

Scratch 專題習作設計工作紙	PJS-2
定義問題	PJS-3
設計你的 Scratch 故事	PJS-5
設計你的 Scratch 故事：分鏡劇本	PJS-6
設計你的 Scratch 遊戲	PJS-7
待辦清單	PJS-9
同儕互評工作紙	PJS-11
小組簡報同儕互評工作紙	PJS-13
反思	PJS-14
小組簡報	PJS-15
重溫基本程序編寫結構的概念	PJS-16
重溫主要概念和實踐	PJS-17
附錄 - 重溫 Scratch 元件	PJS-20

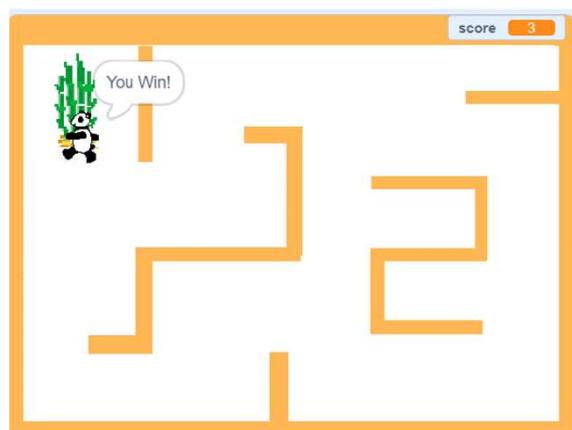
這是你展示編程技能及數碼創意的機會！

- ❑ 你在這個單元將設計及創建一個 Scratch 專題習作。
- ❑ 開始之前，讓我們再玩一遍之前各單元所學的 Scratch 專案，並回顧一下至今你能掌握哪些 Scratch 元件！

可參考附錄 P.20-21



你還記得海底的魚兒、熊貓走出迷宮和 Gobo 與朋友去野餐聊天嗎？



Scratch 專題習作設計工作紙

- 二人一組，設計並創建一個 Scratch 專案。嘗試尋找日常生活中遇到的問題，並透過 Scratch 專案來解決！（例如幫助有需要人士改善生活、廢物回收等）
- 專案可以是遊戲，也可以是故事。先想想你想創建什麼類型專案，你將會有 8 個教節課來設計並完成。

想一想

- 想一想你們的組別想要做什麼類型專案。
- 你想選擇哪個主題？
- 你想運用以前的哪一個 Scratch 專案，並以它為基礎上再構建一些新東西嗎？



定義問題

與你的組員討論一下。構思你的 Scratch 專案主題及設計。
將你的構思記錄在此工作紙上。

1. 你身邊有沒有人遇到一些問題？她 / 他是誰？

2. 他 / 她面臨什麼問題？

3. 你想如何利用 Scratch 來幫助他 / 她去解決這個問題嗎？請描述你的想法。

定義問題

以下範例供參考。

以下是一些例子供你參考。如果你遇到以下情況，你會怎麼做？

祖母想了解科技的發展，但較少機會接觸新科技。



現時有些學生運動量不足，面臨肥胖的問題。你能否創建一個 Scratch 專案，為他們提供一些關於保持健康的小貼士、或設計一個遊戲讓他們多活動身體嗎？



你的同學在考試前感到壓力。你能透過 Scratch 給她講個笑話，讓她放鬆嗎？



你年幼的弟弟不熟悉回收的種類。你可以嘗試為他創建一個學習回收分類的遊戲嗎？



你的小妹妹喜歡數學遊戲。你可以為她使用 Scratch 創建一個數學遊戲嗎？

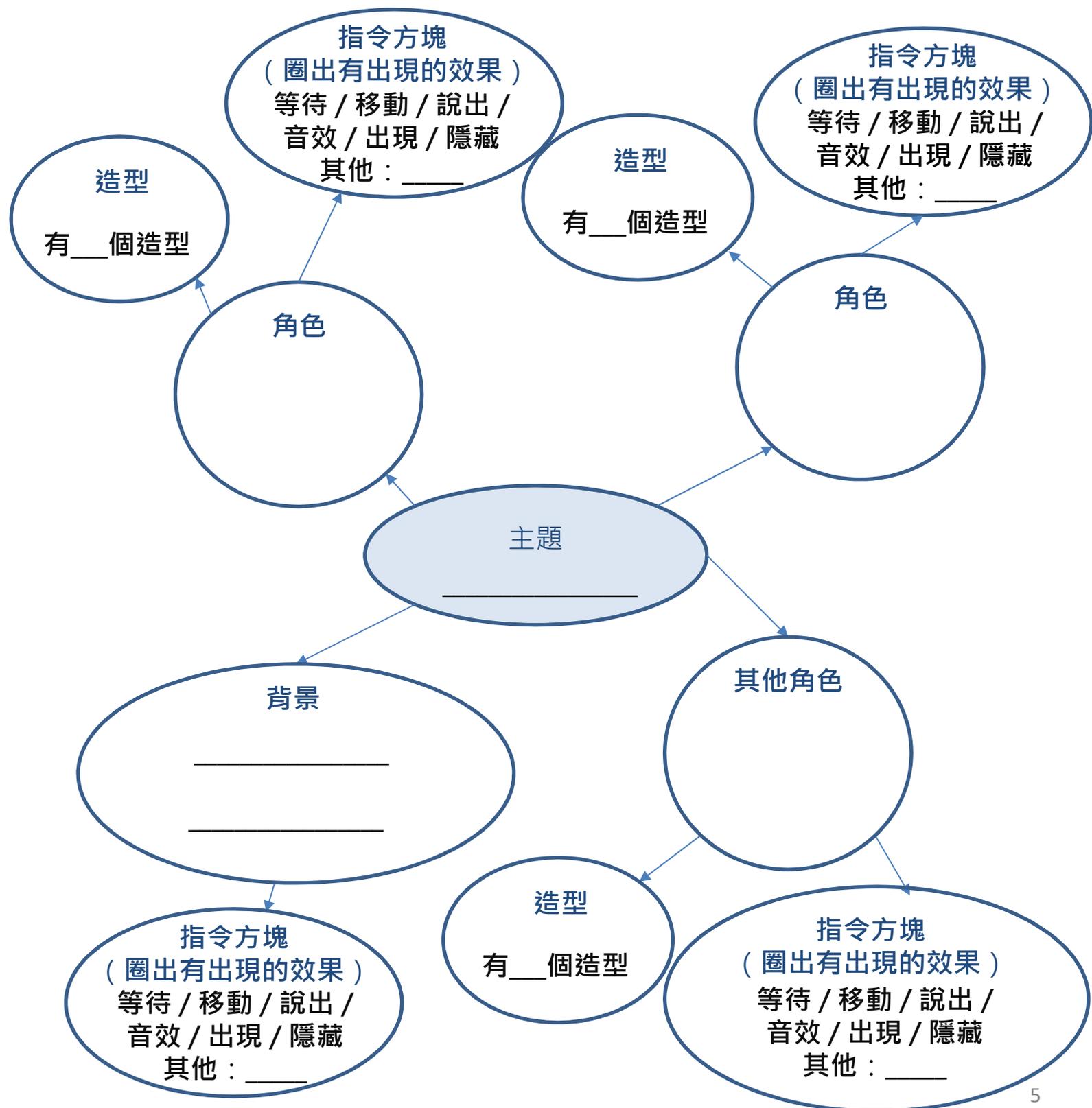


你的朋友喜歡音樂！你能製作一個可以播放不同音樂的音樂盒嗎？



設計你的 Scratch 故事

你打算怎樣用 Scratch 編寫「帶著祖母去旅行」故事？利用以下腦圖列出你故事中的元件。

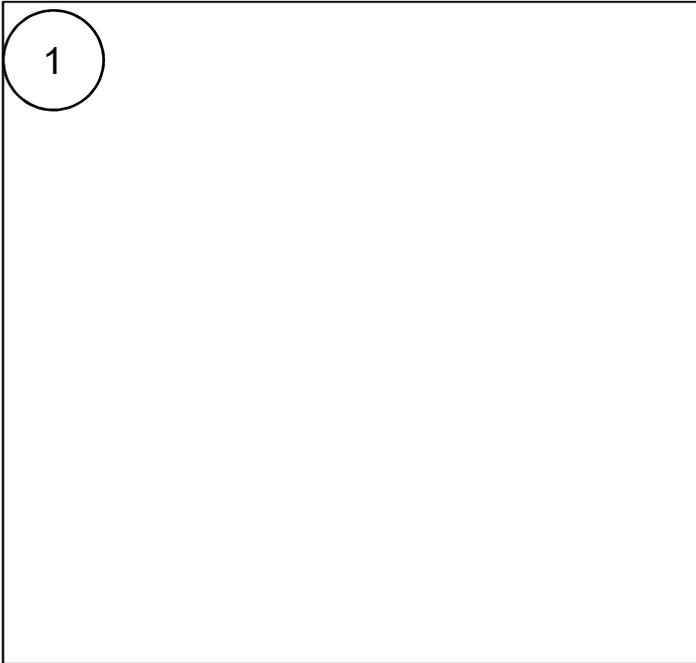
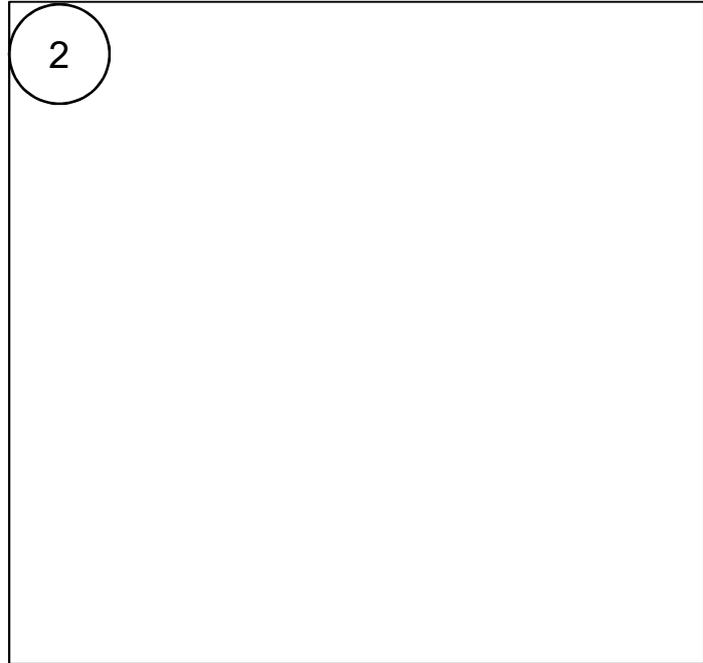
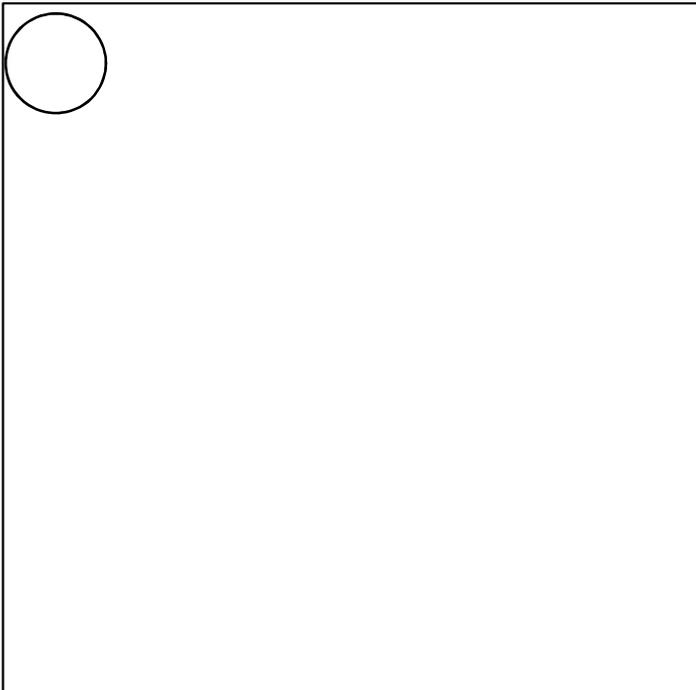
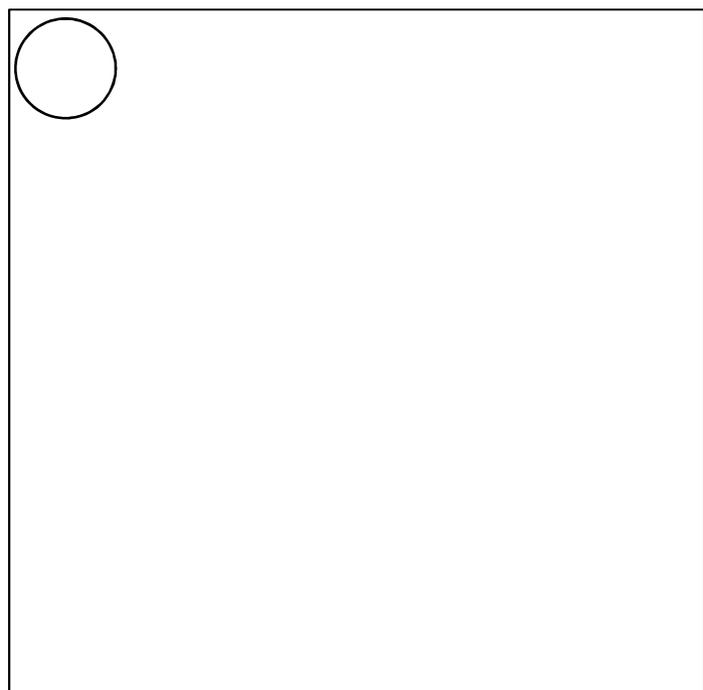


設計你的 Scratch 故事：分鏡劇本

嘗試在以下方格中，按序繪製圖像來設計專案，並將數字 1、2、3 和 4 標記在圓圈內。

試想想，並在空格寫下你的想法。

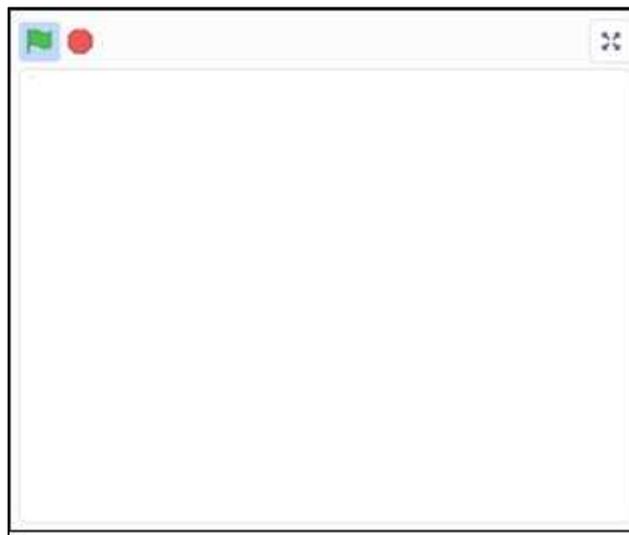
1. 描述故事中不同角色的造型 / 動作？
2. 介紹不同的場景的背景設計

設計你的 Scratch 遊戲

你打算怎樣用 Scratch 編寫遊戲？試在下面表格表達你的想法！

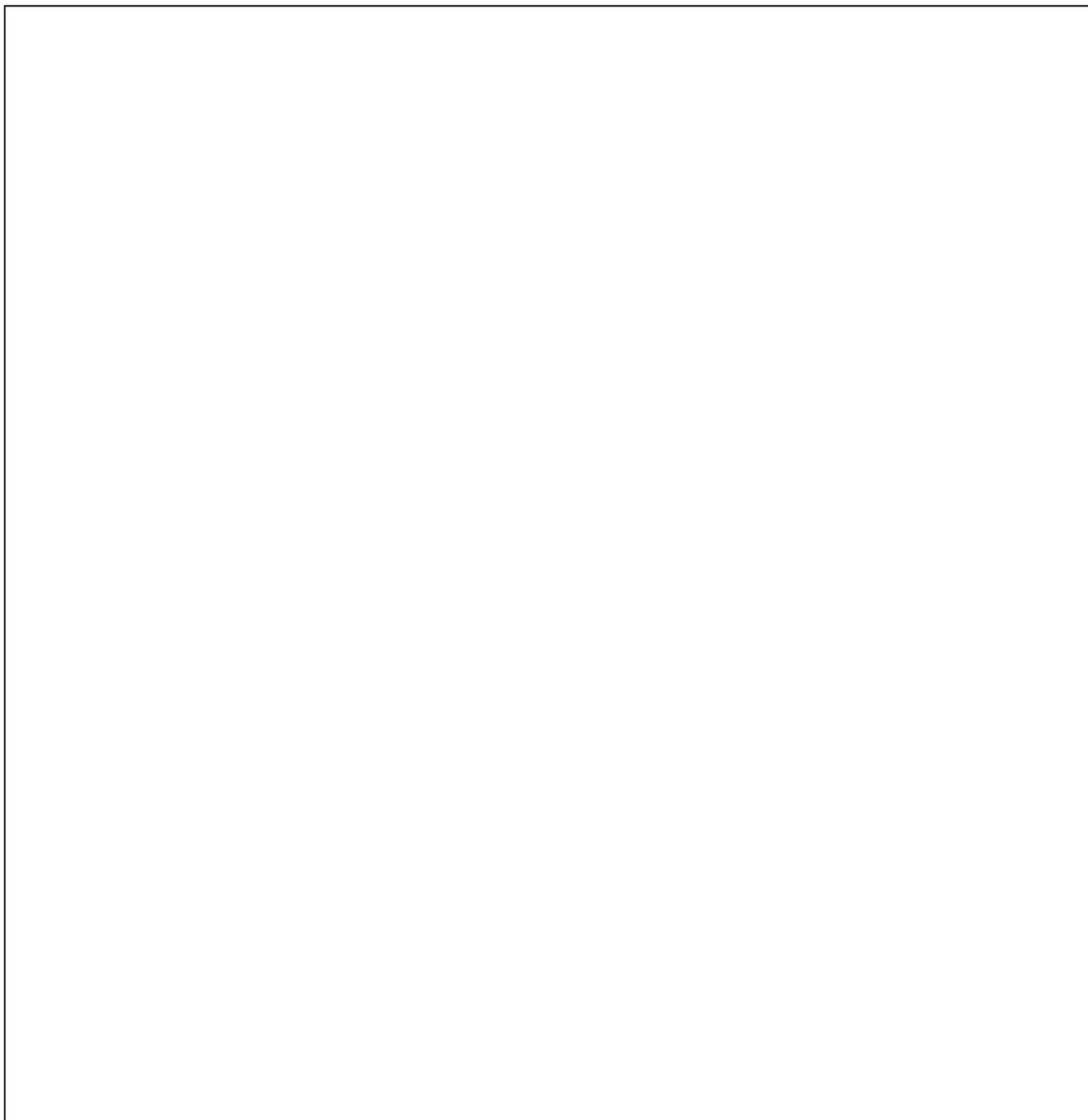
遊戲設計



角色	
動作	
遊戲設計 / 規則	

設計你的 Scratch 遊戲

利用以下流程圖表達你的想法。請留意，遊戲應該有開始和結束。過程中應包含一個條件以作決策。



待辦清單

- 請在開始編程前，制訂待辦清單，這是跟進進度的好方法。
- ❖ 於每教節前，寫下你想完成的事項。
- ❖ 於每教節結束前，寫下已完成的事項或阻礙你完成該任務的原因。

日期 / 教節	要完成的任務	狀態 (已完成 / 遇到問題， 解決方案)

與另一小組互評

同儕互評工作紙

與另一小組互評，仔細聆聽其他組別的簡報。嘗試為他們提供一些具建設性的回饋，以獲得更好的設計！

例如： 「我很喜歡你們**程序的背景**。」  「**角色**生動有趣。」
 「希望**角色**小貓可以在不同的場景轉換**不同的造型**。」
 「請加入**音效**。」

組別 ()

請寫下你喜歡這個專案的兩個部份。





其他建議



組別 ()

請寫下你喜歡這個專案的兩個部份。





其他建議



小組簡報

□ 按照教師的指示，你可以先透過「同儕互評工作紙」改善你的專案，再進行小組簡報。

□ 當你口頭介紹時，請簡介：

1. 自己及組員：

2. 專案的主題 / 目的（發現及解決了什麼問題）：

3. 你最自豪的部分是什麼：

4. 你在專案中克服了哪些困難。

小組簡報

小組簡報同儕互評工作紙

仔細聆聽其他組別的簡報。嘗試為他們提供一些具建設性的回饋，以獲得更好的設計！

例如： 「我很喜歡你們**程序的背景**。」  「**角色**生動有趣。」
 「希望**角色**小貓可以在不同的場景轉換**不同的造型**。」
 「請加入**音效**。」

組別 ()

請寫下你喜歡這個專案的兩個部份。





其他建議



組別 ()

請寫下你喜歡這個專案的兩個部份。





其他建議



反思

自我評估

在方格內剔選 ，並簡單記下重點。

1. 你能否實現專案的目標？

能 我能做得更好

2. 如果你可以重做專案，你將如何改進專案？你會更改方案設計嗎？
你會更改編程的方式嗎？

我想改變設計，因為 _____

我想改變編程，因為 _____

3. 作為這個專案的一份子，你扮演了什麼角色？

組長 編程人員 平面設計師 其他 _____

你完成了什麼工作？

4. 你覺得你是個好的組員嗎？你在專題研習中有沒有尊重和鼓勵你的組員嗎？

是 / 有 我能做得更好

因為我..... _____

小組簡報

分享到創作坊以作小組匯報

當你完成專案，請將專案添加到教師的創作坊。

1. 點擊「檔案」內的「儲存」選項，儲存你的專案。
2. 點擊橙色的「分享」按鈕。



3. 進入教師的工作室（你的教師會給你一個網址）。
4. 你可以在「添加專案」一欄中添加你的專案，通過「網址添加」或「瀏覽專案」完成添加。



5. 你如果選擇「瀏覽專案」，你會看到所有已分享的專案。按下「+」把合適的專案加入創作坊。



重溫基本程序編寫結構的概念

當你擬訂專案時，想想你以前學過的基本程序編寫結構、主要概念和實踐及各種的 Scratch 指令方塊。嘗試用不同的指令方塊編寫你的程序。

序列

程序的序列是指執行編程指令的次序。



分支 / 選擇 - 條件

可以透過「如果 - 那麼」指令方塊來決定角色的動作。



循環 / 重複

「重複無限次」和「重複」可引發重覆的動作。



重溫主要概念和實踐

事件

運用事件來控制角色，例如角色定位。



更換造型

通過改變角色的造型來展示動作！



變數

變數可以幫助你在 Scratch 專案中儲存和更新數值。



廣播

廣播和當收到訊息指令方塊可讓角色相互交談，並觸發動作。



命名

給角色起個合適名字也很重要。



重溫主要概念和實踐

函式積木

如果你使用自訂的函式積木，可將複雜的任務分解為更小並可重複使用的指令方塊。



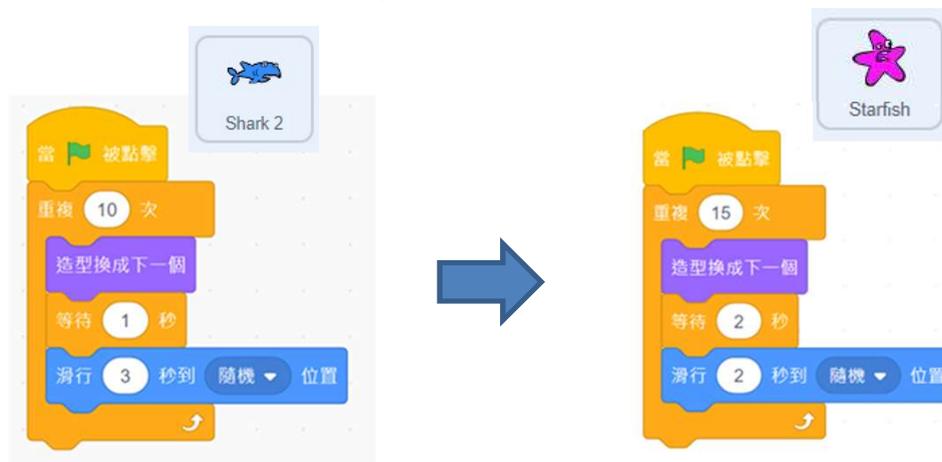
廣播

廣播和當收到訊息指令方塊可讓角色相互交談，並觸發動作。



重用及混合程序 / 編碼

我們可以重用和混
合一個角色的編碼，
應用到第二個及第
三個角色。



重溫主要概念和實踐

條件邏輯運算符

我們用邏輯運算符判斷一個條件是真還是假。可運用比較運算符 (> 、 < 及 =) 及邏輯運算符 (AND 、 OR 、 NOT) 來表達條件。



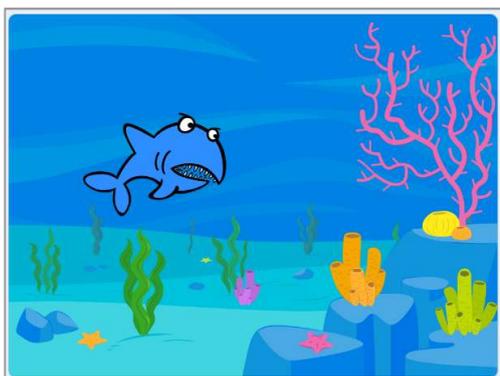
重溫 Scratch 元件

- 讓我們回顧一下你之前在 Scratch 單元中學到了哪些元件！你可隨時再次播放之前所學的專案及進行重溫！

單元

學到的重要元件

單元 2



<https://scratch.mit.edu/projects/722781437>



單元 3



<https://scratch.mit.edu/projects/761138953/>



單元 4



<https://scratch.mit.edu/projects/727401089>

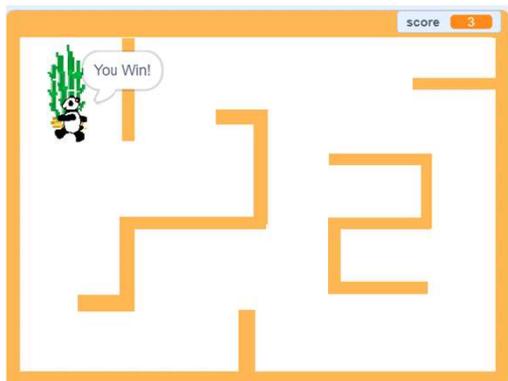


重溫 Scratch 元件

單元

學到的主要元件

單元 5



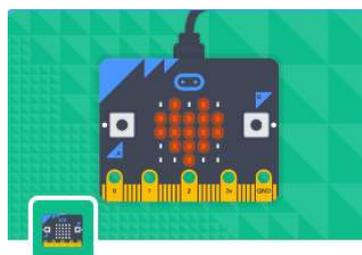
<https://scratch.mit.edu/projects/722154863>



單元 6



<https://scratch.mit.edu/projects/734787236/>



micro:bit

讓你的專案與實體世界連結。

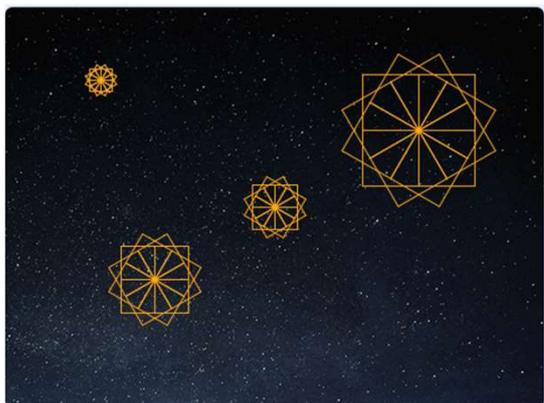
需求



合作者

micro:bit

單元 7、8



<https://scratch.mit.edu/projects/737985288>



