

Enriched Module on Coding Education for Upper Primary Level – The Design and Pedagogy

高小增潤編程教育課程單元-課程單元 設計及教學法

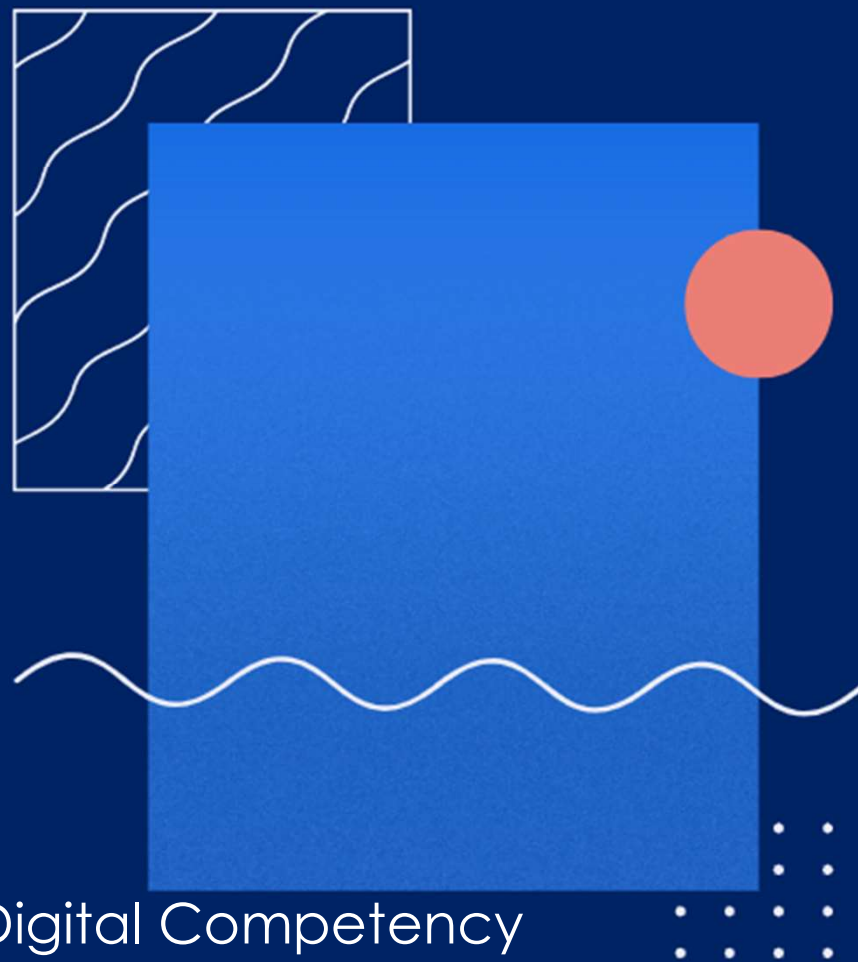
香港教育大學 江紹祥教授

Professor Kong Siu-Cheung,

Research Chair Professor of E-Learning and Digital Competency

The Education University of Hong Kong

16 June 2023 二零二三年六月十六日



目標

學生將能夠：

- 明白計算思維的基本概念與實踐，包括抽象化、算法和自動化。
- 具備開發程序及數據處理的能力以解決問題。
- 瞭解解決問題的過程和編程的局限性。
- 將編程與現實生活中的問題和其他科目連繫起來。
- 在過程中透過溝通及有效的團隊合作以解決問題。

計算思維—編程教育

小學課程補充文件

課程發展議會編訂

香港特別行政區政府教育局建議學校採用

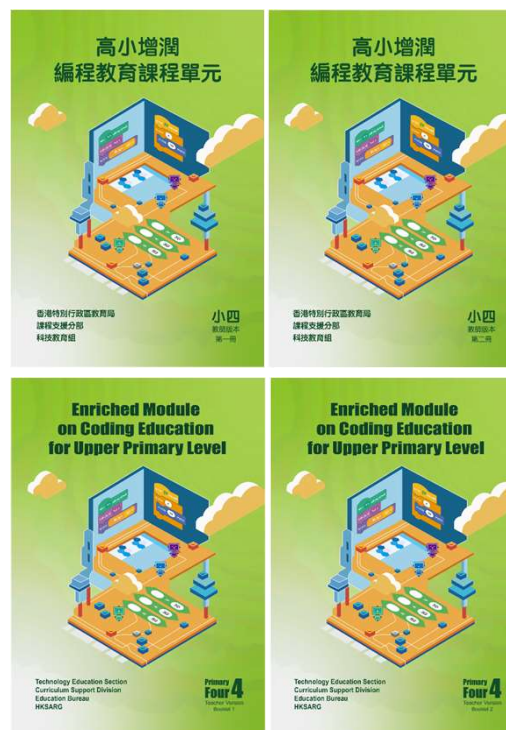
二零二零

在高小年級推行計算思維和編程教育，目的並非訓練及培養電腦程序編寫員，而是讓學生得到實作經驗及建立解難的信心，透過協作及重覆的測試來解決問題。

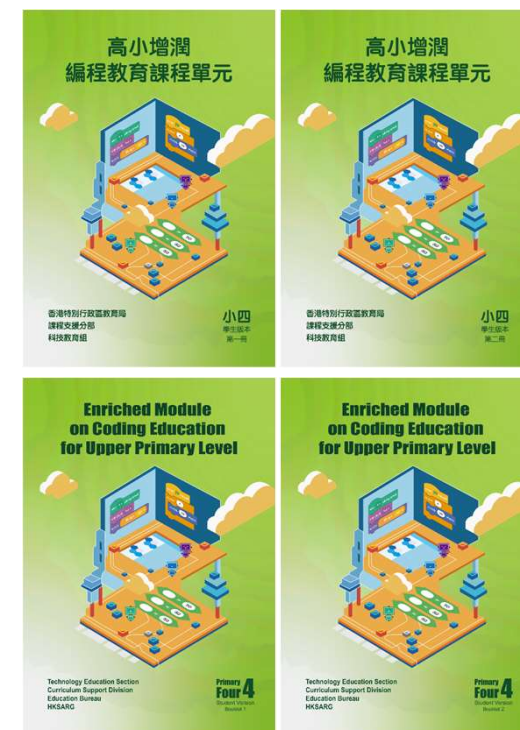
高小增潤編程教育課程單元

- 高小增潤編程教育課程設計緊貼2020年公布更新的《計算思維 - 編程教育：小學課程補充文件》
- 建議課時:14小時 (6個核心單元及專題習作)
- 2個可供選擇的延伸單元：供學校為能力較高及對編程有興趣的學生，加深他們對編程及創新科技的理解

教師版(中文及英文)

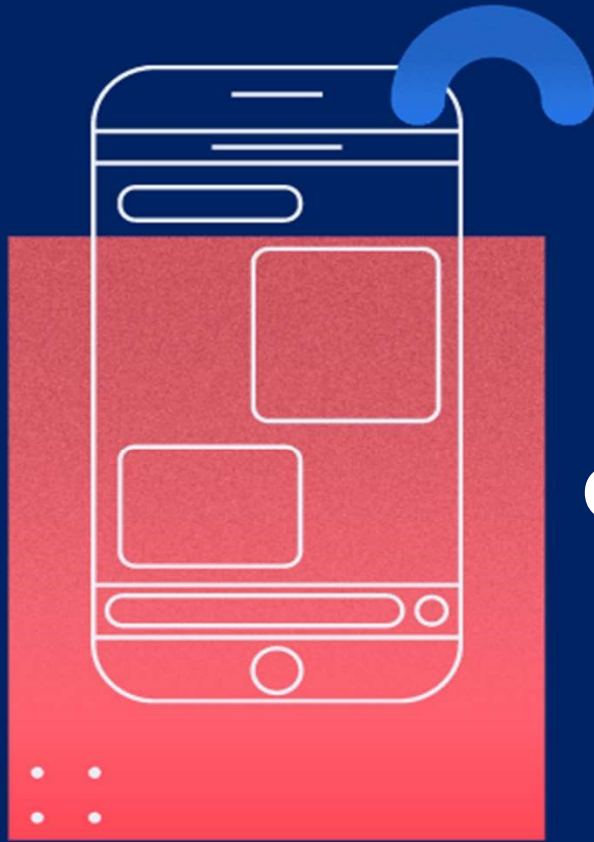


學生版(中文及英文)



高小增潤編程教育課程單元

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



Computational Thinking Development

計算思維發展

計算思維 = 編程教育？

Computational Thinking = Coding Education?

- What coding education wants to achieve?
推動編程教育的目的是什麼？
- What computational thinking development wants to achieve?
發展計算思維的目的是什麼？

計算思維 = 編程教育？

Computational Thinking = Coding Education?

- What coding education wants to achieve?
- 推動編程教育的目的是什麼？

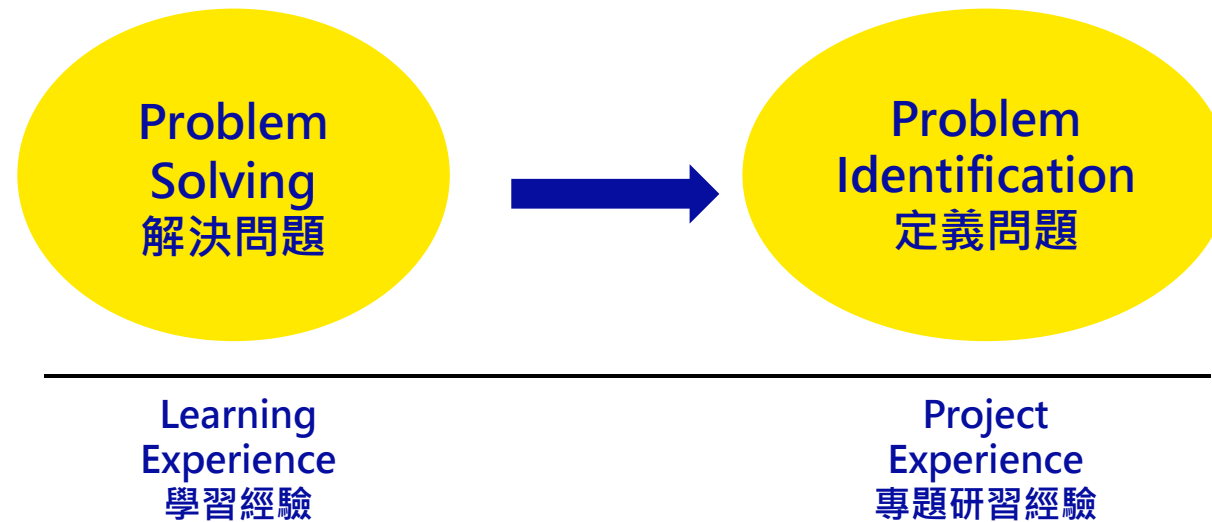
To develop computational thinking
發展計算思維

- What computational thinking development wants to achieve?
- 發展計算思維的目的是什麼？

To nurture problem-solvers with digital creativity
培育具數碼創意的解難人才

Nurture Students' Problem-solving Skills and Digital Creativity

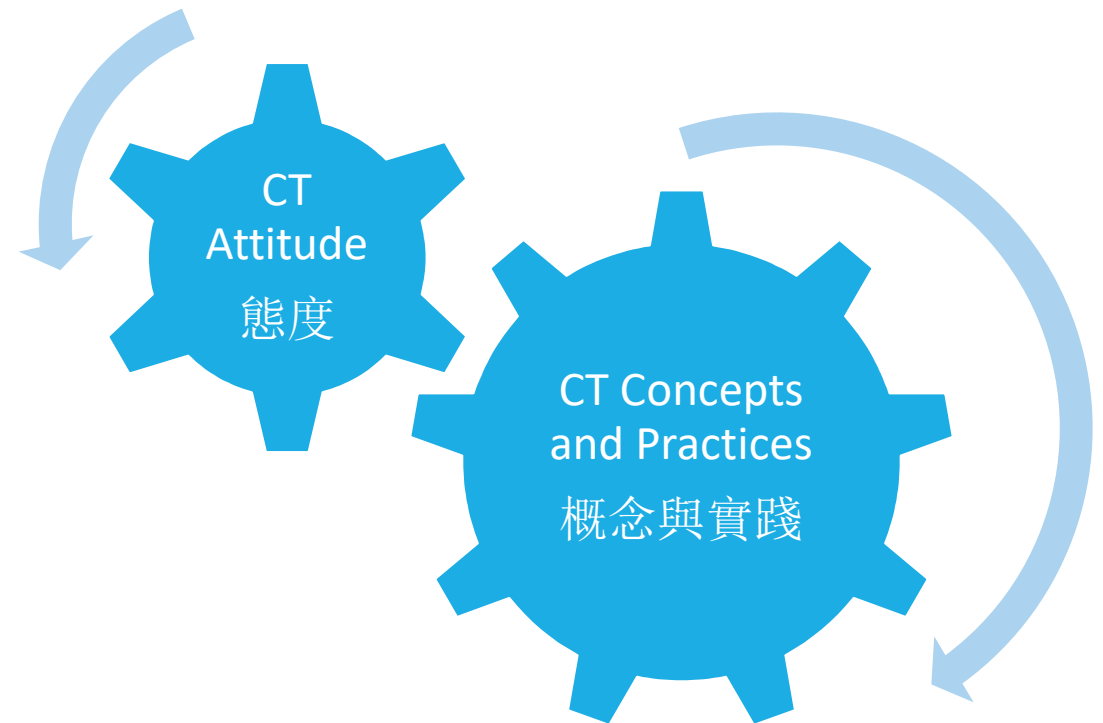
培養學生的解難能力和數碼創意



Computational Thinking 計算思維

- 計算思維是每個人於數碼世界中，具有**解難 (problem solver)**、**創新 (innovative creator)**及**批判思考 (critical thinker)** 的基礎能力。
- 它包括**定義問題 (problem identification)**，**解決問題 (problem solving)**，**系統設計 (system design)**，以及透過電腦科學的基本概念理解人類行為。

(EDB, 2020; Wing, 2006)



Core of Computational Thinking 計算思維的核心

Problem identification 定義問題

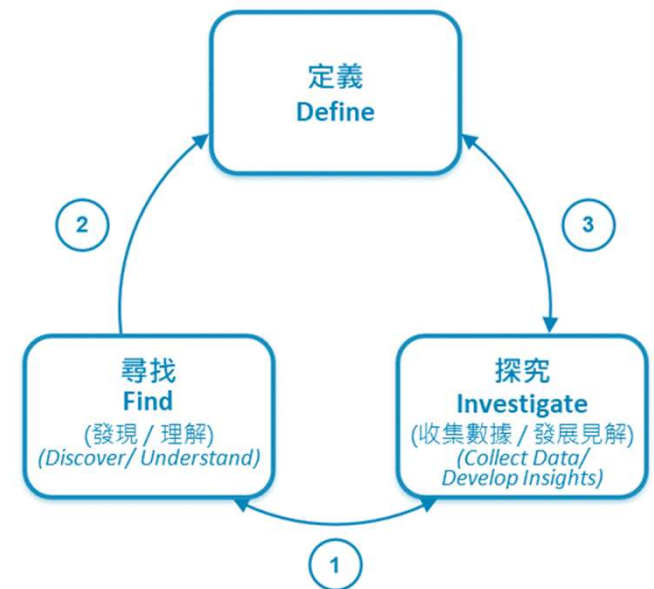
The ability to discover problems in the real life or raise questions in the digital world.

日常生活中發掘問題或在數碼世界中提出問題

Problem solving 解決問題

The ability to systematically process information and design the algorithm to solve problems.

有系統地處理資訊及設計算法來解決問題



(Kong, 2022)

數碼充權 Digital Empowerment

編程活動對自己的
認知價值
Perceived value of
Programming activities

意義性
(Meaningfulness)

影響
(Impact)

認為可透過掌握編
程來解決世界問題
Perception that one can
solve the world's problems
by mastering programming

相信具備必要的技術及
能力來完成編程工作
Belief that one has the
necessary skills and abilities to
perform a programming task
well

編程自我效能表
現
(Programming
Self-efficacy)

創意自我效能
(Creative
Self-efficacy)

相信能夠創建創新想法
及方案
Belief that one can produce novel
ideas and solutions

(Kong, Chiu & Lai, 2018)

Key Foci to be Achieved through Coding Education

編程教育的目標

To nurture problem-solvers with digital creativity

培育具數碼創意的解難人才

Pedagogical design to develop problem-solving skills
and inspire digital creativity

利用教學設計發展解難能力，激發數碼創意

To Play, To Think, To Code, To Reflect, To Create

玩一玩，想一想，來編程，齊反思，同創作



Seven-Step Guide of TPACK in CTE

計算思維教育 (CTE) 的科技教學學科知識(TPACK)七步曲



什麼是科技教學學科知識 (TPACK) ?

範疇	意思
CK	Content Knowledge (內容知識)
PK	Pedagogical Knowledge (教學知識)
PCK	Pedagogical Content Knowledge (教學內容知識)
TK	Technology Knowledge (科技知識)
TCK	Technological Content Knowledge (科技內容知識)
TPK	Technological Pedagogical Knowledge (科技教學知識)
TPACK	Technological Pedagogical Content Knowledge (科技教學學科知識)

計算思維教育(CTE)的科技教學學科知識 (TPACK) 七步曲

第一步: TCK (Scratch)

第二步: CK (計算思維概念、實踐和態度)

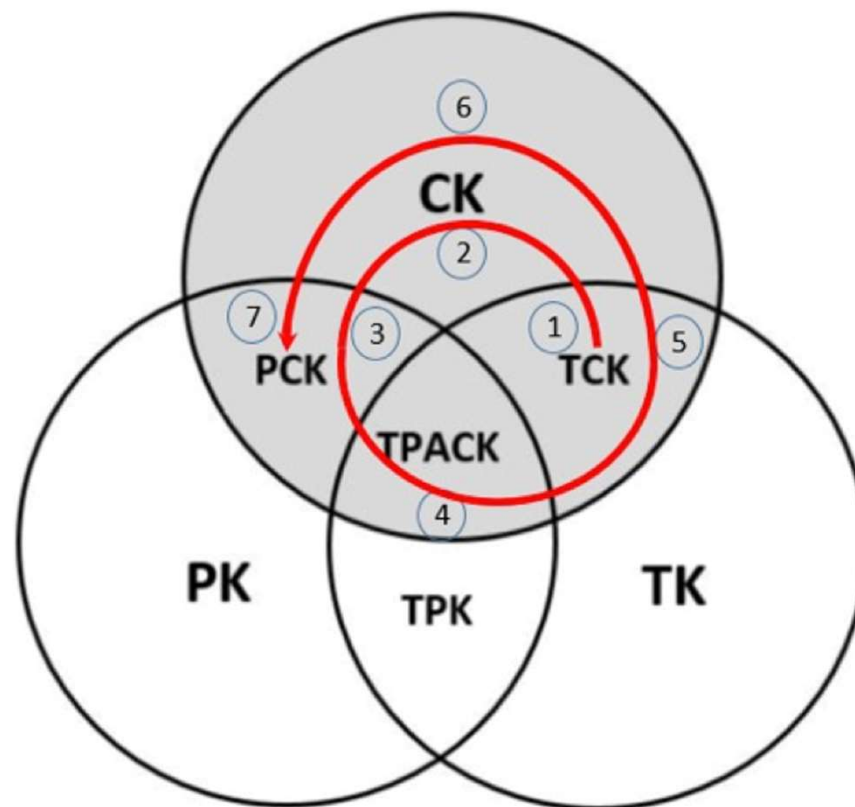
第三步: PCK (教學內容知識)

第四步: TPACK (科技教學學科知識)

第五步: TCK (數碼創意)

第六步: CK (總結計算思維概念、實踐和態度)

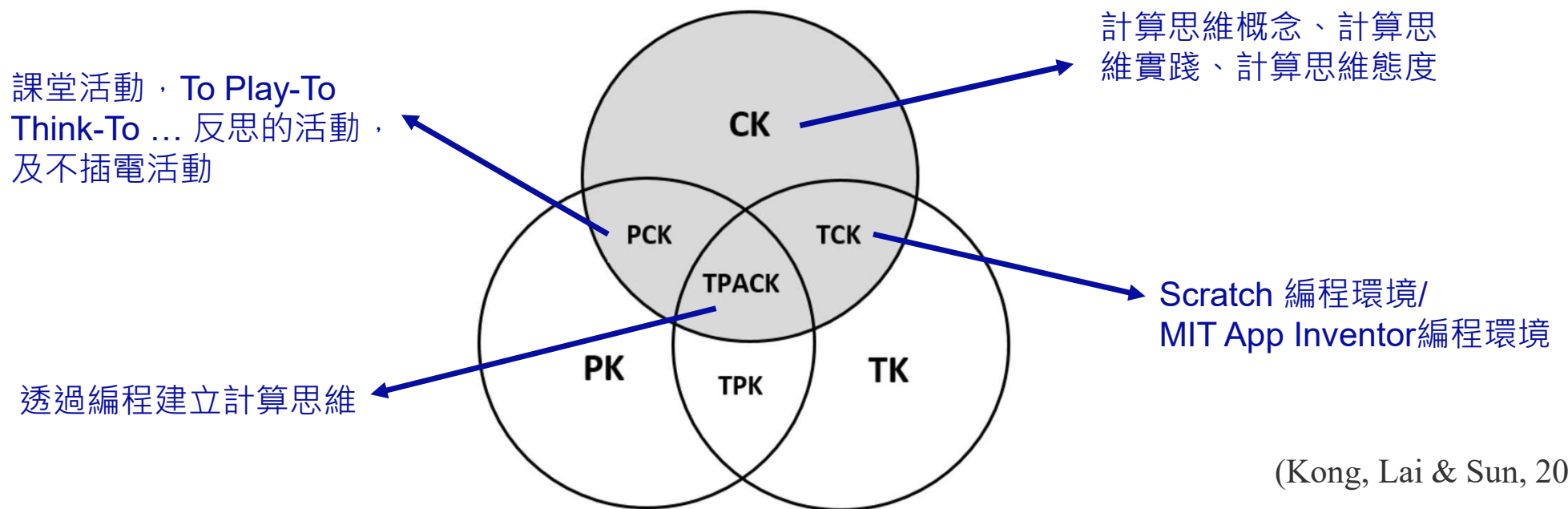
第七步: PCK (教師自我反思教學法)



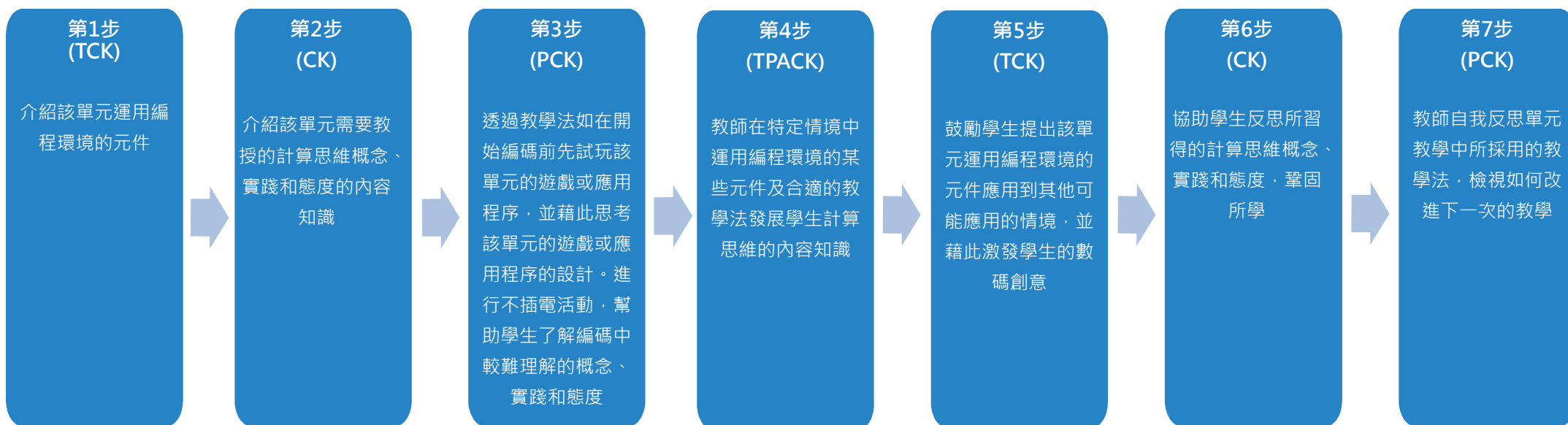
(Kong, Lai & Sun, 2020)

計算思維教育(CTE)的科技教學學科知識 (TPACK) 七步曲

內容知識 (CK), 科技內容知識 (TCK), 教學內容知識 (PCK) 及科技教學學科知識 (TPACK) 是在計算思維教育裏教師專業培訓的重要範疇。



計算思維教育(CTE)的科技教學學科知識 (TPACK) 七步曲



Teachers can flexibly arrange the steps to groom students' problem-solving skills and digital creativity.
教師可以彈性安排教學步驟，以發展學生的解難能力和數碼創意。



Seven-Step Guide of TPACK in Each Unit

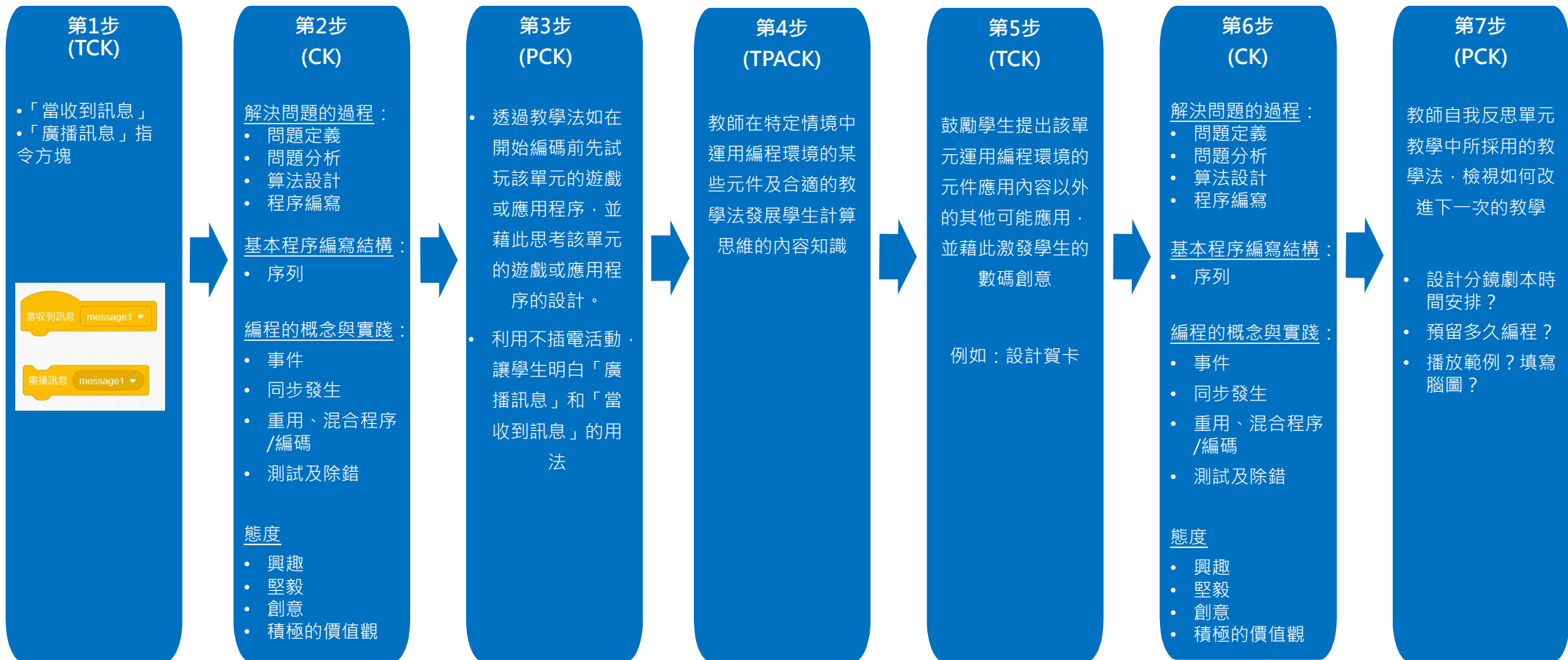
科技教學學科知識(TPACK)七步曲在各個單元的運用



單元示例：說故事 Storytelling



單元示例：說故事 — TPACK七步曲的運用



(1) Scratch 編程環境： 介紹「廣播訊息」和「當收到訊息」指令方塊



(2) 學習目標

1. 透過創作專案，展示對「廣播訊息」、「當收到訊息」指令方塊及切換背景的理解。
2. 將計算思維概念和實踐中的序列，應用到Scratch專案製作中；重用及混合一個已有的專案及其素材。
3. 通過學生創作自己的故事，並與同儕、教師和家長分享，來培養學生的創意。

(2) 計算思維概念和實踐

主要學習元素	項目
抽象化	把問題拆解為小問題
算法	<p><u>解決問題的過程</u>： 問題定義、問題分析、算法設計、編程</p> <p><u>基本程序編寫結構</u>： 序列</p> <p><u>編程的概念與實踐</u>： 重用、混合程序 / 編碼，測試及除錯</p>

(2) 計算思維態度

1. 培養學生對編程的興趣；
2. 鼓勵學生在測試及除錯中表現出堅毅及積極的態度；
3. 鼓勵學生展示創意及創造力以創作他們的故事；
4. 通過說故事活動培養學生正面的價值觀和態度（例如：關注環境保護議題）。

(2) 計算思維概念和實踐

事件



同步發生



(2) 計算思維概念和實踐

重用及混合程序 / 編碼：

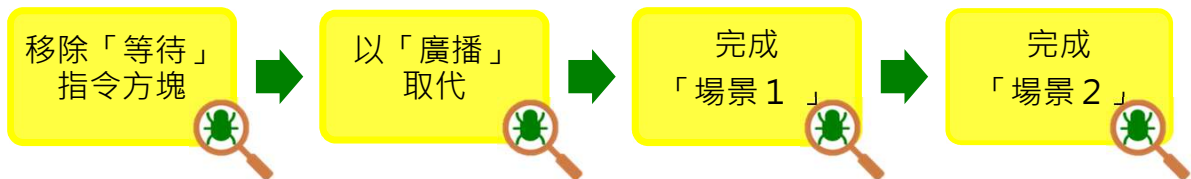


(2) 計算思維概念和實踐

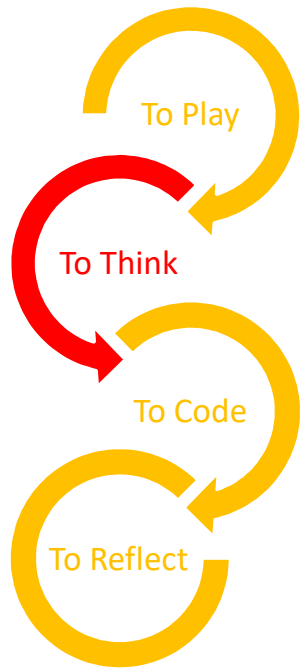
反覆構思及漸進編程



測試與除錯



(3) 想一想 (To Think)



想一想 To Think:

- 這裏有多少個角色？
- 他們說了些什麼？
- 如何營造角色交談的效果？
- 對話的速度會否太快或太慢？你如何調整對話的速度？

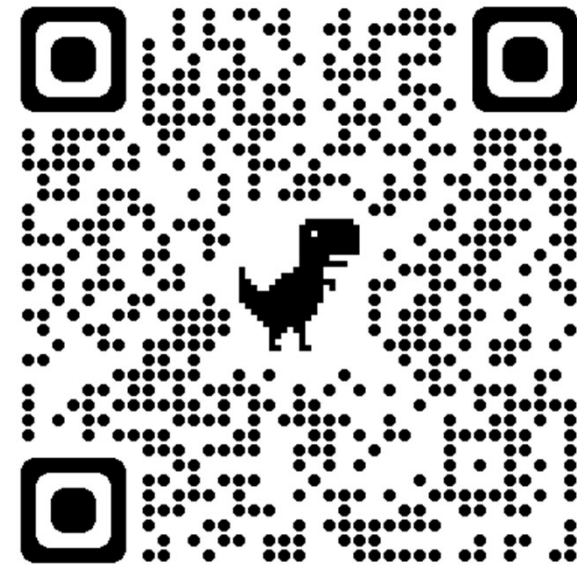
「玩一玩」時，留意
以下問題...

(3) 玩一玩 (To Play)

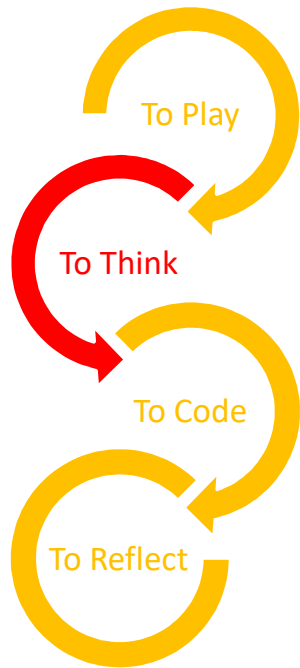
玩一玩 To Play:

- 播放故事 (範例)

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



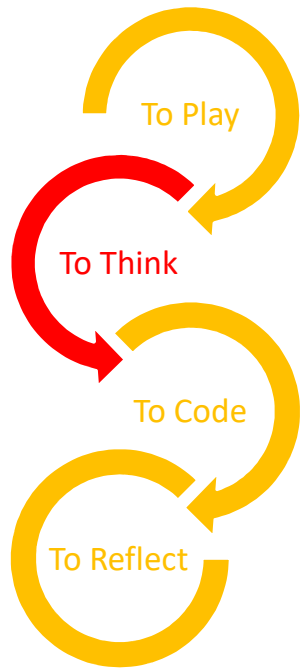
(3) 想一想 (To Think)



想一想 To Think:

- 這裏有多少個角色？
- 他們說了些什麼？
- 如何營造角色交談的效果？
- 對話的速度會否太快或太慢？你如何調整對話的速度？

(3) 想一想 (To Think)

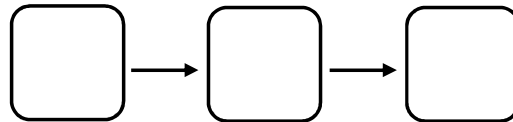


想一想 To Think:

- 這裏有多少個會說話角色？(2個)
- 他們說了些什麼？(野餐, 環保)
- 如何營造角色交談的效果？(於編程時自行探索答案)
- 對話的速度會否太快或太慢？你如何調整對話的速度？(於編程時自行探索答案)

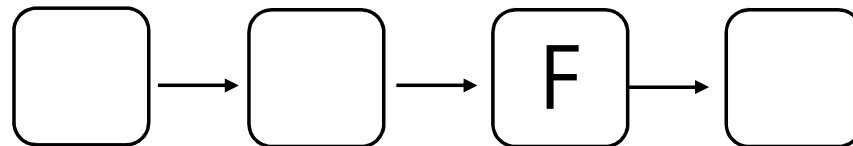
(3) 想一想(To Think)

「場景 1」車站 (Metro)



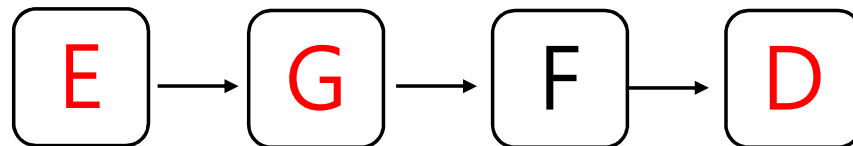
(3) 想一想 (To Think)

「場景 2」 – 海邊 (Boardwalk)



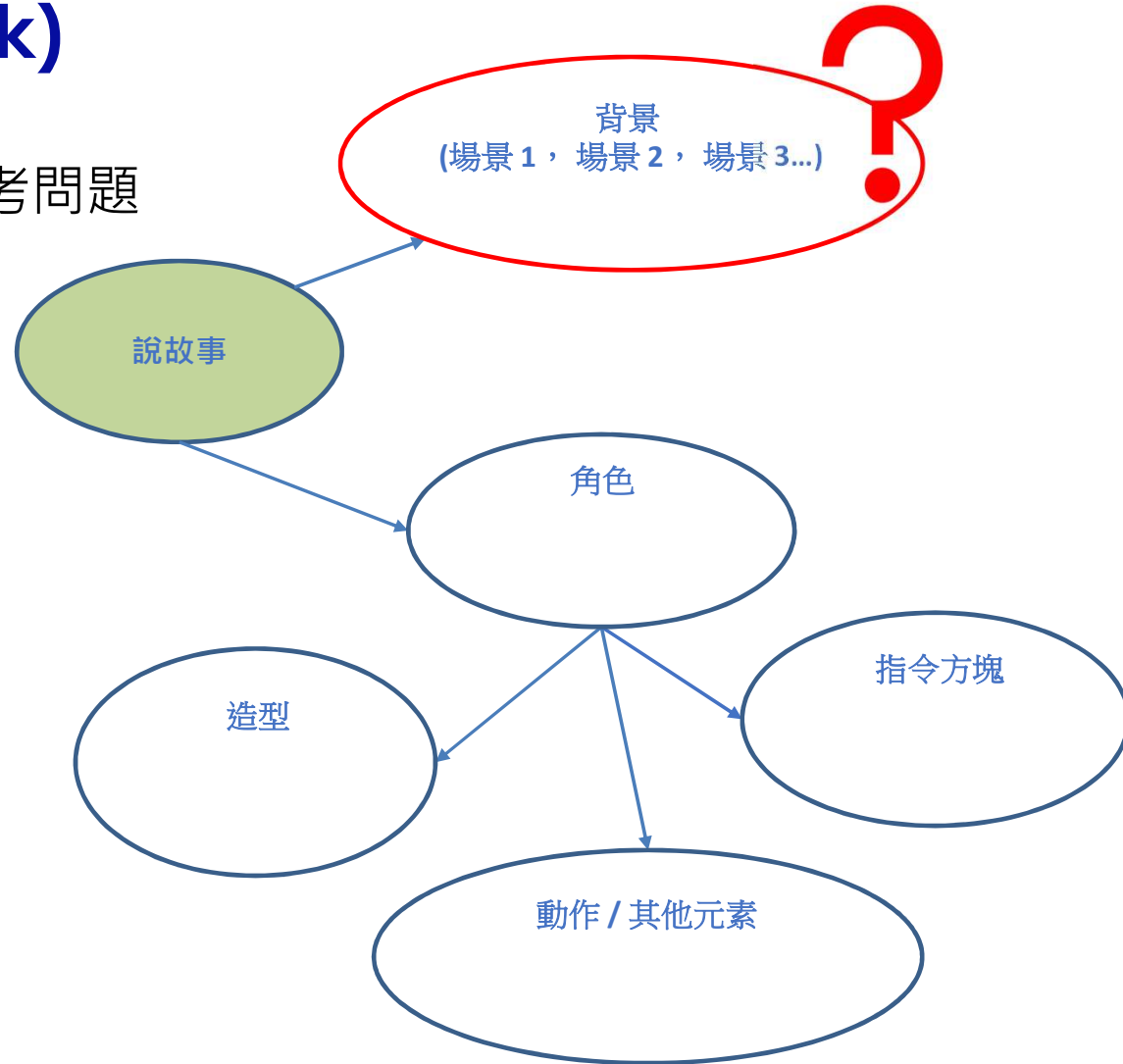
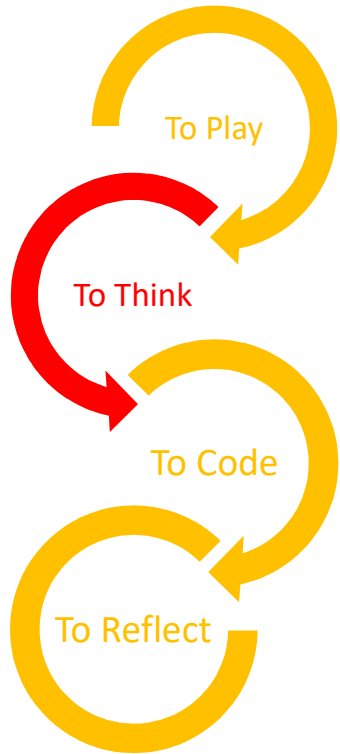
(3) 想一想 (To Think)

「場景 2」 – 海邊 (Boardwalk)



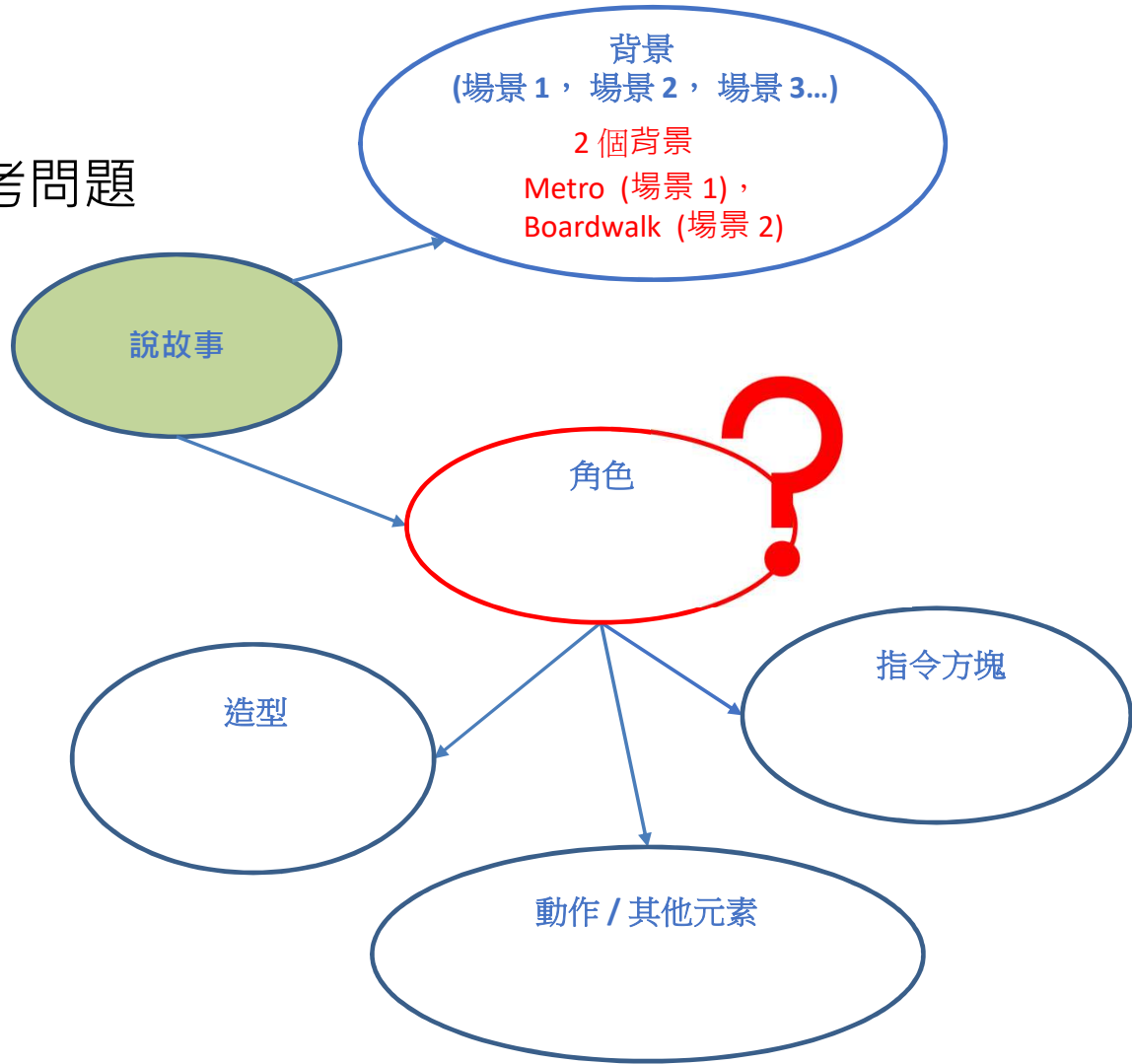
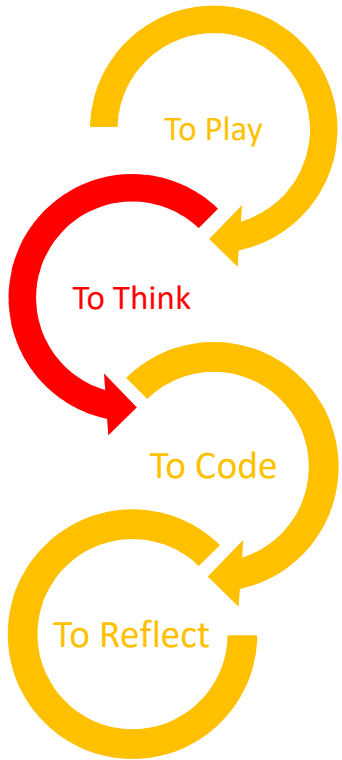
(3) 想一想 (To Think)

利用腦圖思考問題



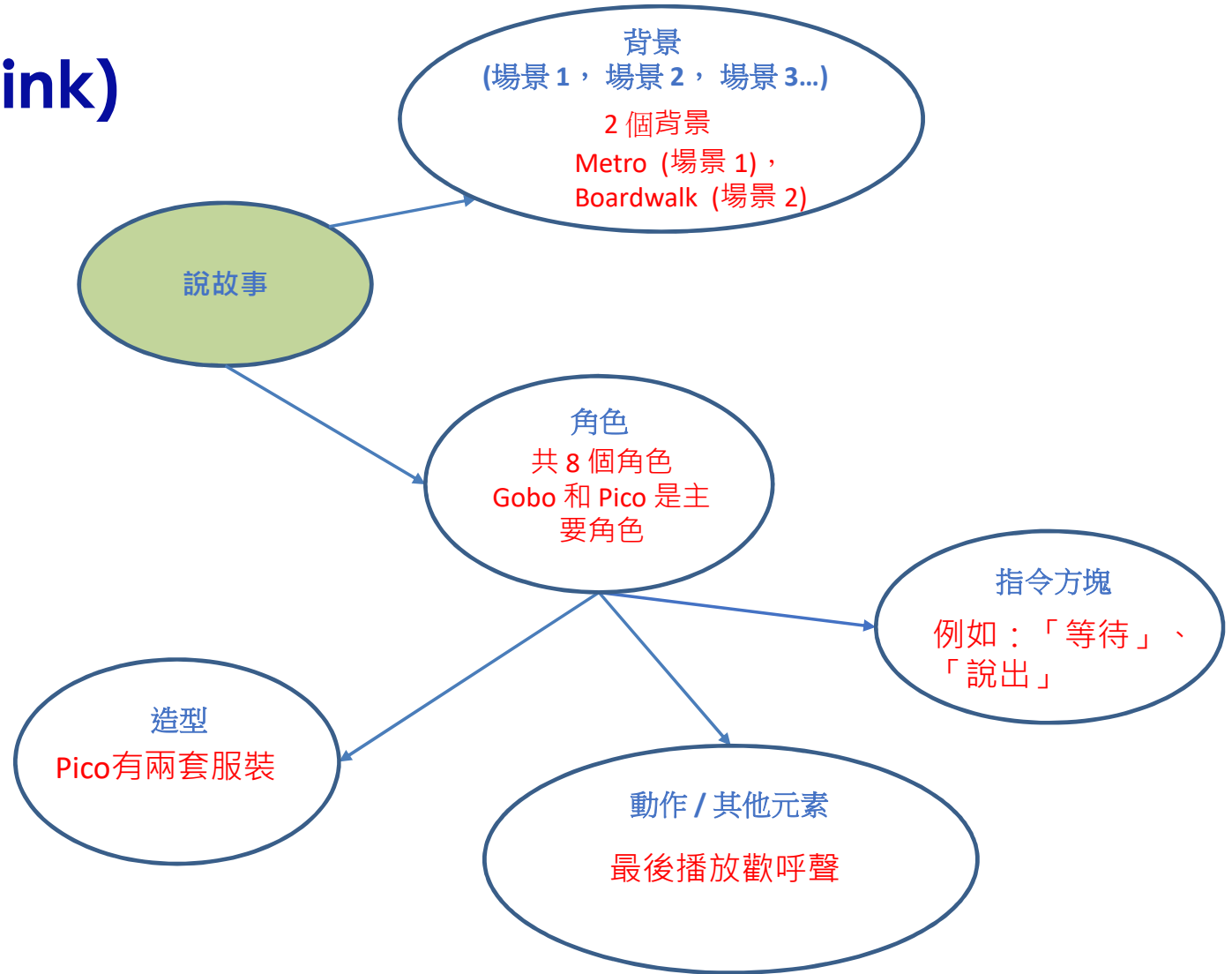
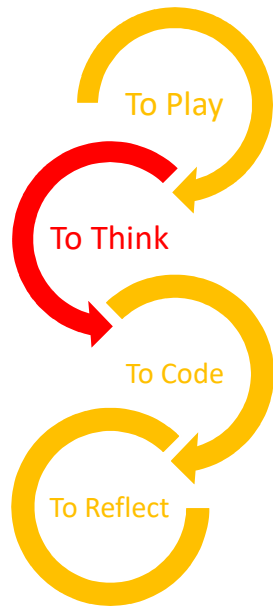
(3) 想一想 (To Think)

利用腦圖思考問題

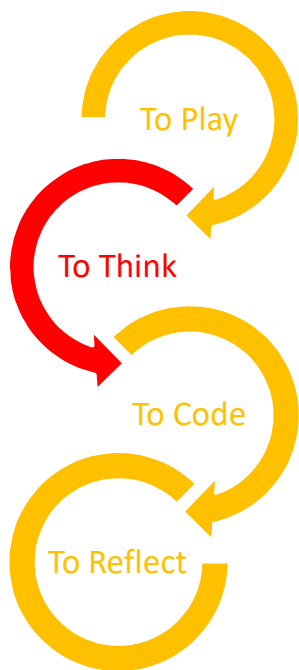


(3) 想一想 (To Think)

利用腦圖思考問題



(3) 學一學(To Learn)



利用不插電活動
讓學生明白「廣播訊息」和
「當收到訊息」的用法

不插電課堂活動：廣播的角色扮演

本次角色扮演活動將邀請兩名學生扮演 Gobo 和 Pico 的角色，展示 Gobo 在說話後如何向 Pico 發出信號。活動目的在於讓學生明白「廣播訊息」和「當收到訊息」的用法。

學生 1 (Gobo) 	學生 2 (Pico) 
當點擊綠旗 時 說：我們乘計程車去野營吧！ 說：「廣播訊息 1」	
	當我收到訊息 1 說：我們乘巴士吧，它更環保。 說：「廣播訊息 2」
當我收到訊息 2 說：好的，我們走吧！ 說：「廣播訊息 3」	
	當我收到訊息 3 說：為什麼剩下這麼多食物？這樣太浪費了。 說：「廣播訊息 4」
當我收到訊息 4 說：但我已經吃飽了…… 說：「廣播訊息 5」	
	當我收到訊息 5 說：那我們下次不要點太多食物，避免浪費資源。 說：「廣播訊息 6」
當我收到訊息 6 說：我們都是環保戰士！	當我收到訊息 6 說：我們都是環保戰士！

(4) 科技教學學科知識 (TPACK)

請在 <https://scratch.mit.edu/projects/761138953>
進入Storytelling的專案

移除「等待」指令方塊

來編程：移除「等待」指令方塊

現在我們將使用「廣播訊息」及「當收到訊息」的指令方塊來取代「等待」指令方塊。請按情節提要的每個部份分拆 Gobo 和 Pico 的編碼，然後刪除「等待」指令方塊。

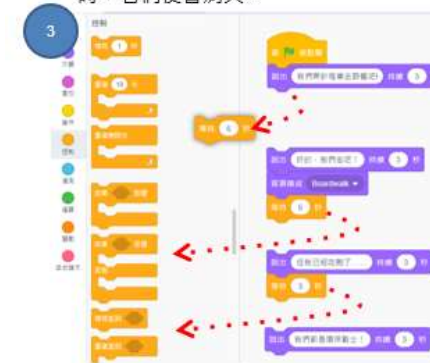
1. 點擊「角色」以打開 Gobo 的講稿。



2. 分拆指令方塊。



3. 將「等待」指令方塊拖出編程區域時，它們便會消失。



4. 完成！



(4) 科技教學學科知識 (TPACK)

以「廣播訊息」代替「等待」指令方塊

來編程：以「廣播訊息」及「當收到訊息」代替

請使用「廣播訊息」及「當收到訊息」的指令方塊取代所有在角色對話中使用的「等待」指令方塊。請留意這兩個指令方塊都在事件抽屜中。

- 從 Gobo 在「場景 1」中的第一句對白開始，並在末端添加一塊「廣播訊息 message1」指令方塊，而訊息的編號需要遵循分鐘的順序。



- 切換到 Pico 的編碼，並在她第一句對白頂部加上「當收到訊息1」指令方塊。



(4) 科技教學學科知識 (TPACK)

完成「場景 1」的第二組對話

來編程：完成「場景 1」的第二組對話

請為所有「場景 1」的對白加上「廣播訊息」指令方塊。現在 Gobo 需要向 Pico 廣播她的回應，輪到 Gobo 發言了。

1. 拖出一個新的「廣播訊息」指令方塊，並點擊 message1，再選取 New message。
2. 將它命名為 message2。



3. 將「廣播訊息」的指令方塊移到 Pico 說話指令方塊的末端。



(4) 科技教學學科知識 (TPACK)

完成「場景 2」的對話

來編程：完成「場景 2」的對話

現在，讓我們繼續在海邊的「場景 2」。他們去了野餐！

- 分鏡的下一步 (4) 是 Pico 的下一句對話。
- 請為 Pico 的講稿加上「當收到訊息」和「廣播」的指令方塊，並把新的訊息命名為 message4。



場景 1

插圖	Gobo	Pico
	<p>說出 我們設計程車去野嘅吧! 持續 3 秒</p> <p>廣播訊息: message1</p>	
		<p>接收訊息: message1</p> <p>說出 我們係巴士呢 持續 3 秒</p> <p>說出 它更環保 持續 3 秒</p> <p>廣播訊息: message2</p>
	<p>接收訊息: message2</p> <p>說出 好的, 我們走吧! 持續 3 秒</p> <p>廣播訊息: message3</p> <p>背景轉換: Boardwalk</p>	

「場景 2」

插圖	Gobo	Pico
		<p>接收訊息: message3</p> <p>背景轉換: Pico-d</p> <p>說出 點解咁多嘢咁多呢? 持續 3 秒</p> <p>說出 這樣太浪費了 持續 3 秒</p> <p>廣播訊息: message4</p>
	<p>接收訊息: message4</p> <p>說出 你已經吃飽了? 持續 3 秒</p> <p>廣播訊息: message5</p>	
		<p>接收訊息: message5</p> <p>背景轉換: Pico-d</p> <p>說出 唔好! 你唔係咁多人食嘢, 啱食好幾多! 持續 3 秒</p> <p>廣播訊息: message6</p>
	<p>接收訊息: message6</p> <p>說出 我們都是環保嘅士! 持續 3 秒</p> <p>播放音效: Cheer</p>	<p>接收訊息: message6</p> <p>說出 我們都是環保嘅士! 持續 3 秒</p> <p>播放音效: Cheer</p>

(4) 科技教學學科知識 (TPACK)

同創作：分鏡劇本

是時候來設計一個屬於你的故事了！





故事主題：

- 環保
- 智慧城市
- 關愛家人等相關議題

1	

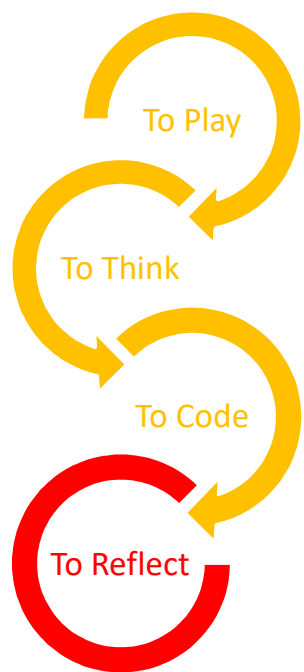
參考例子

前往: <https://scratch.mit.edu/projects/775613221/>

場景	Sasha	Characters 2
		
		



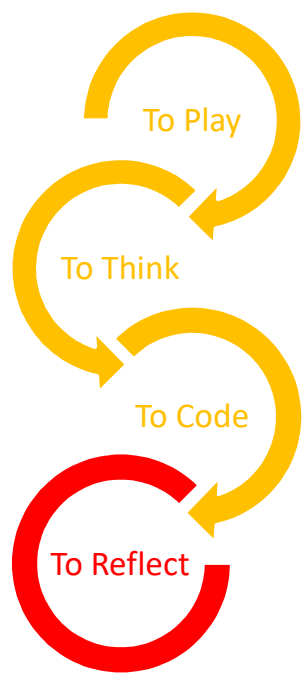
(5) 科技內容知識反思，啟發數碼創意



❖ 你能想到有應用「廣播訊息」功能的現實例子嗎？



(5) 科技內容知識反思，啟發數碼創意



❖ 你能想到有什麼現實例子，是應用了「廣播訊息」功能嗎？

(e.g. 廣播颱風即將來臨；不同訊號的預防訊息)

❖ 你又會運用「廣播訊息」及「當收到訊息」到哪些情境？

(用來創作一個動畫故事，設計賀卡等)



(6) 總結計算思維主要概念和實踐

序列：這是編程的重要概念。程式的序列是指執行編程指令的次序。錯誤的次序會使程序無法正確執行。



事件：我們用事件指令方塊來觸發 Scratch 程序的執行。



同步發生：Scratch 允許多於一個事件在同一時間發生。例如，在 Scratch 的編程裏，我們能設定多個角色在同一時間執行動作。



(6) 總結計算思維主要概念和實踐

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角色（如單元二的鯊魚）的編碼，應用到第二個及第三個角色。



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。



測試與除錯：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。為一個程式除錯就是為程式找出錯誤的源頭並改正錯誤。



(6) 總結計算思維主要概念和實踐

序列 (Sequence)

- 什麼能令故事大綱的對話按次序顯示?



答案:

按照「廣播訊息」和「當收到訊息」指令方塊的“message”序列
例如廣播訊息 message1 > 當收訊息 message1, 如此類推

(6) 總結計算思維主要概念和實踐

事件 (Events)

- 除了綠旗  這事件，還有其他事件嗎？



答案:

「當收到訊息」和「廣播訊息」

(6) 總結計算思維主要概念和實踐

同步發生 (Parallelism)

- 如何令兩個角色同時運行?



答案:

(當「收到訊息」message 6, 兩個角色會根據自己的指令方塊來運行)

(6) 總結計算思維主要概念和實踐

重用及混合 (Reusing and Remixing)

- 編程時怎樣應用重用及混合改進程序/編碼？



答案:

更改「等待」為「廣播訊息」
以改善對話的流暢度

(6) 總結計算思維主要概念和實踐

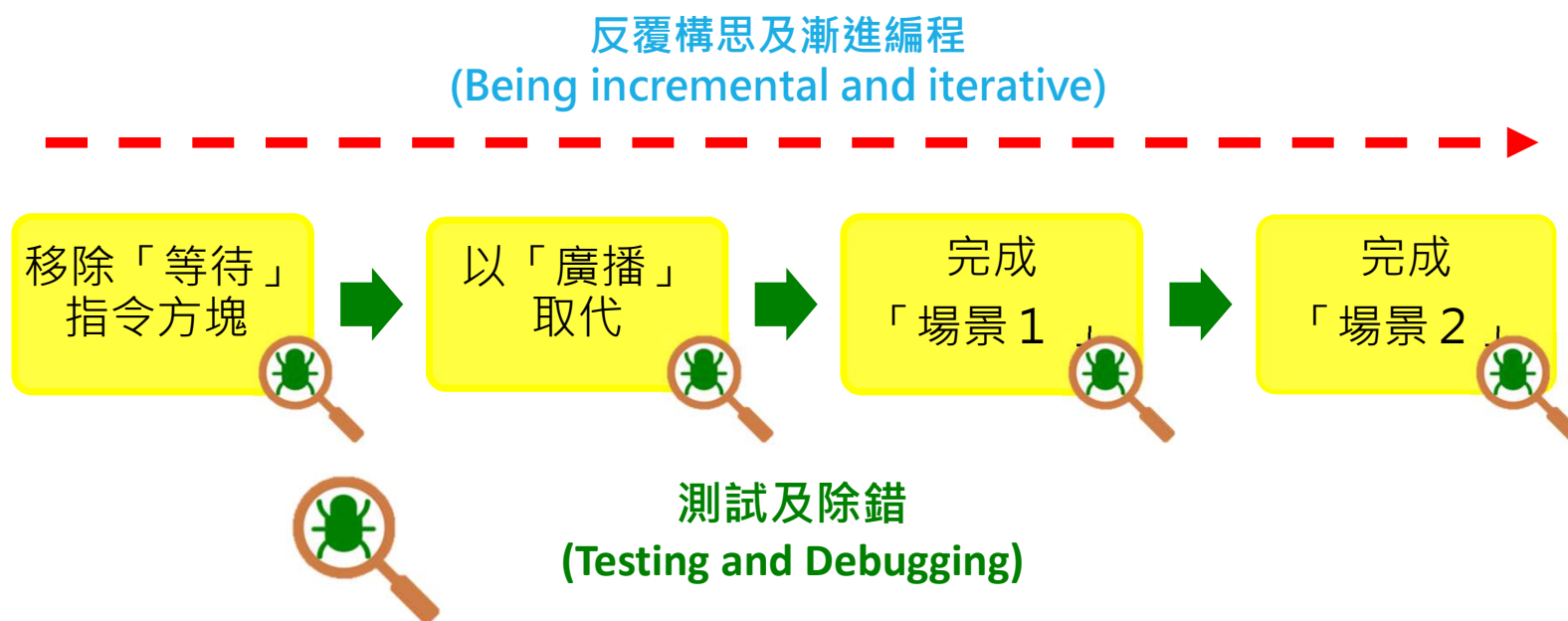
反覆構思及漸進編程

測試及除錯

你怎樣完成這個專案？

答案:

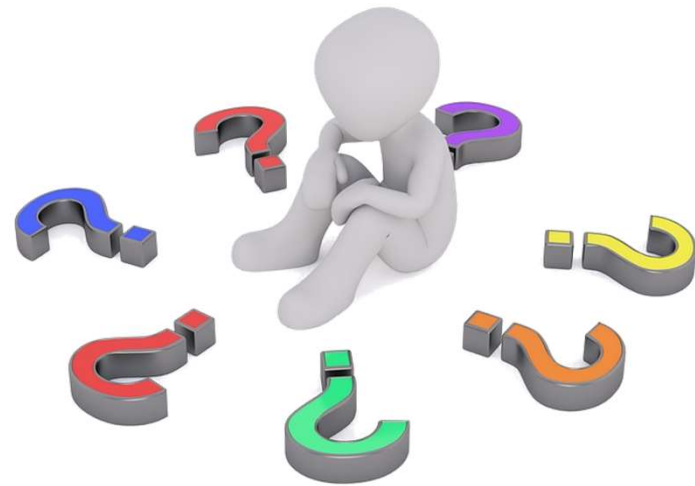
編程時不斷測試及除錯，完成一步後，根據上一步建立了的基礎來改進，直至完成編程。

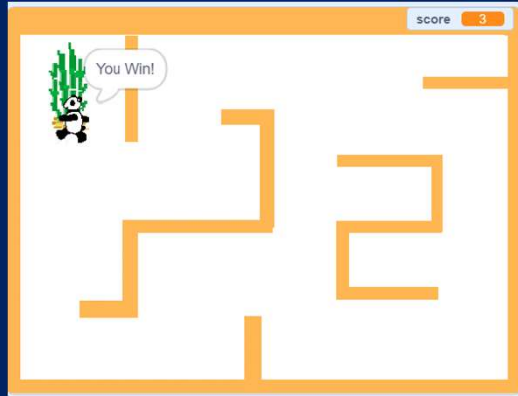


(7) 教師自我反思教學法

如何改善下一次的教學？

- 設計分鏡劇本時間安排？
- 預留多少時間編程？
- 播放範例？填寫腦圖？

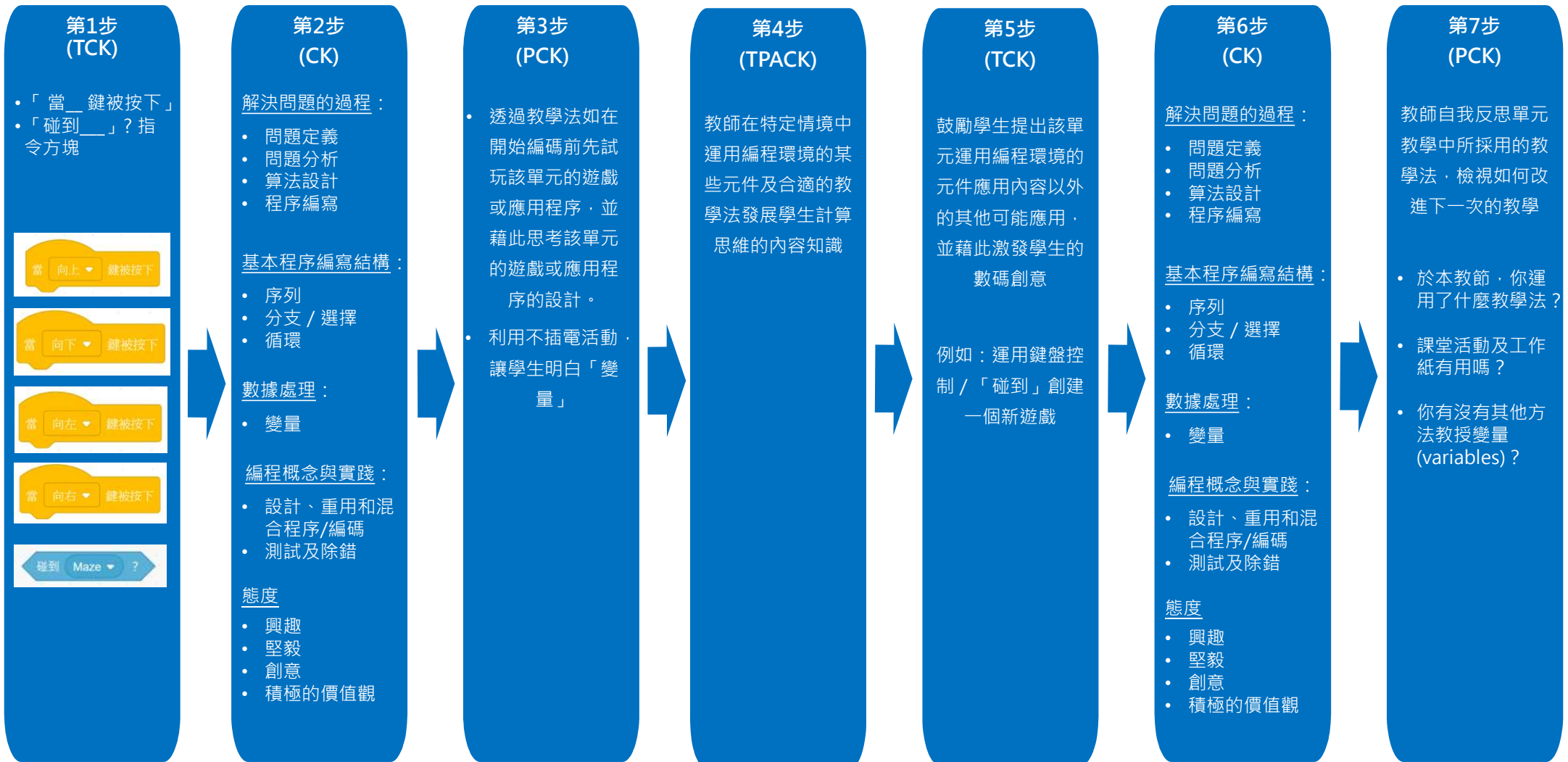




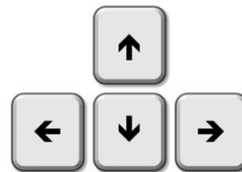
單元示例：創作迷宮遊戲

Make a Maze Game with Scratch

單元示例：創作迷宮遊戲 — TPACK七步曲的運用



(1)Scratch 編程環境：「當__鍵被按下」指令方塊 (事件)



(1) Scratch 編程環境：「碰到___」？指令方塊



(2) 學習目標

1. 學習運用Scratch的鍵盤事件控制角色的移動；
2. 重用和混合編碼轉換角色造型和移動角色的座標；
3. 通過在專案中使用「如果—那麼」的條件句式，來展示學生對該指令方塊的理解；
4. 學生在Scratch專案中適當地使用「重複無限次」指令方塊來重複特定動作，展示對「循環」的理解；
5. 學生在Scratch專案中正確使用碰撞事件；
6. 學生在Scratch專案中，展示對使用變數儲存信息的理解；
7. 通過與同學和教師分享他們的迷宮遊戲，展示編程可以成為一種有趣的社交活動。

(2) 計算思維主要概念和實踐

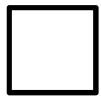
主要學習元素	項目
抽象化	表達算法
算法	<p><u>解決問題的過程</u>： 問題定義、問題分析、算法設計、程序編寫</p> <p><u>基本程序編寫結構</u>： 序列、分支 / 選擇、循環</p> <p><u>數據處理</u>： 變量</p> <p><u>編程概念與實踐</u>： 設計、重用、混合程序 / 編碼、測試及除錯</p>

(2) 計算思維態度

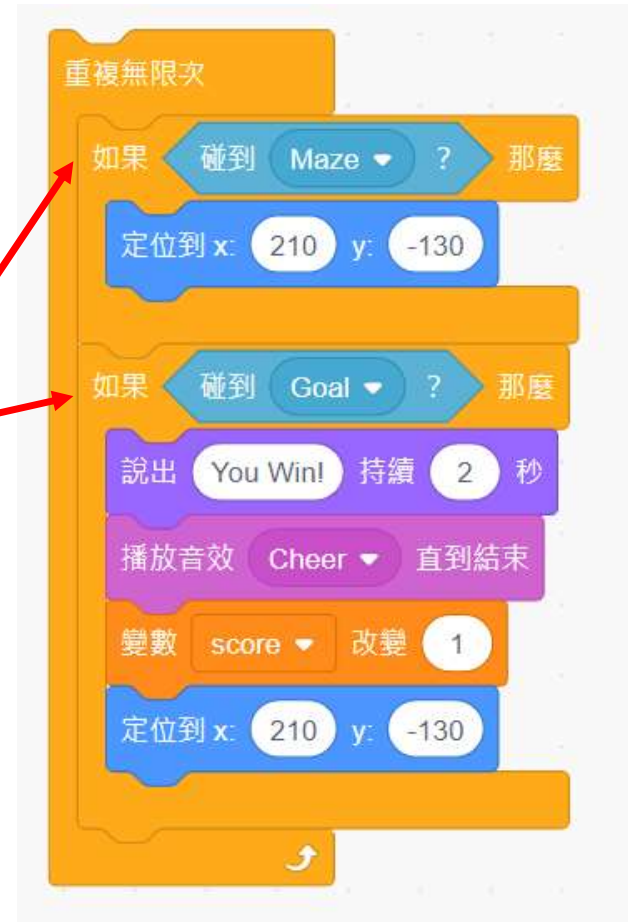
1. 培養學生對編程的興趣；
2. 鼓勵學生在測試及除錯的過程表現出堅毅及積極的態度；
3. 鼓勵學生展示創意及創造力以改進他們的迷宮遊戲。

(2) 計算思維主要概念和實踐

1.



2.



(2) 計算思維主要概念和實踐



The diagram shows a Scratch code block starting with a square sprite labeled '1.' and a red arrow pointing to a script block. The script block consists of three stacked blocks: a yellow '當 向右 鍵被按下' (When the right arrow key is pressed) block, a blue 'x 改變 10' (Change x by 10) block, and a purple '造型換成下一個' (Switch to next costume) block.

圖 1 運用了什麼概念?

這是變數

這是事件

提交

(2) 計算思維主要概念和實踐



The image shows a Scratch code block labeled "重複無限次" (Repeat indefinitely). Inside the loop, there are two "如果" (If) blocks. The first "如果" block has a "碰到 Maze" (Hit Maze) event and a "那麼" (Then) block containing "定位到 x: 210 y: -130" (Go to x: 210 y: -130). The second "如果" block has a "碰到 Goal" (Hit Goal) event and a "那麼" (Then) block containing "說出 You Win! 持續 2 秒" (Say You Win! for 2 seconds), "播放音效 Cheer 直到結束" (Play sound effect Cheer until finished), "變數 score 改變 1" (Change variable score by 1), and "定位到 x: 210 y: -130" (Go to x: 210 y: -130). A red box with the number "2." and a square symbol is positioned to the left of the code block, with two red arrows pointing to the "碰到 Maze" and "碰到 Goal" event blocks.

2.

圖 2 運用了什麼概念?

這是條件

這是循環 - 重複無限次

提交

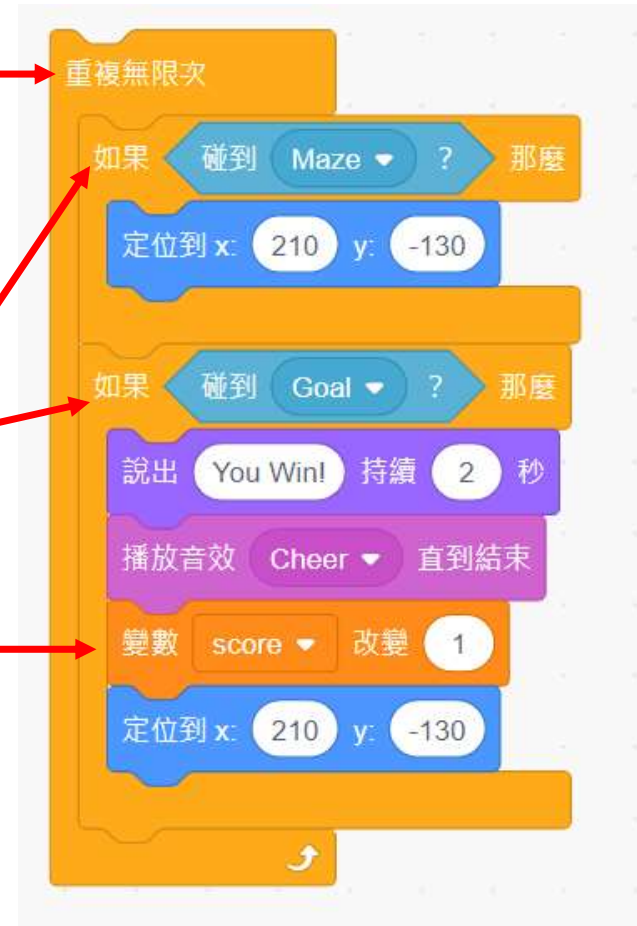
(2) 計算思維主要概念和實踐



循環 (Iteration) –
重複無限次 (Forever)

條件 (Conditionals)

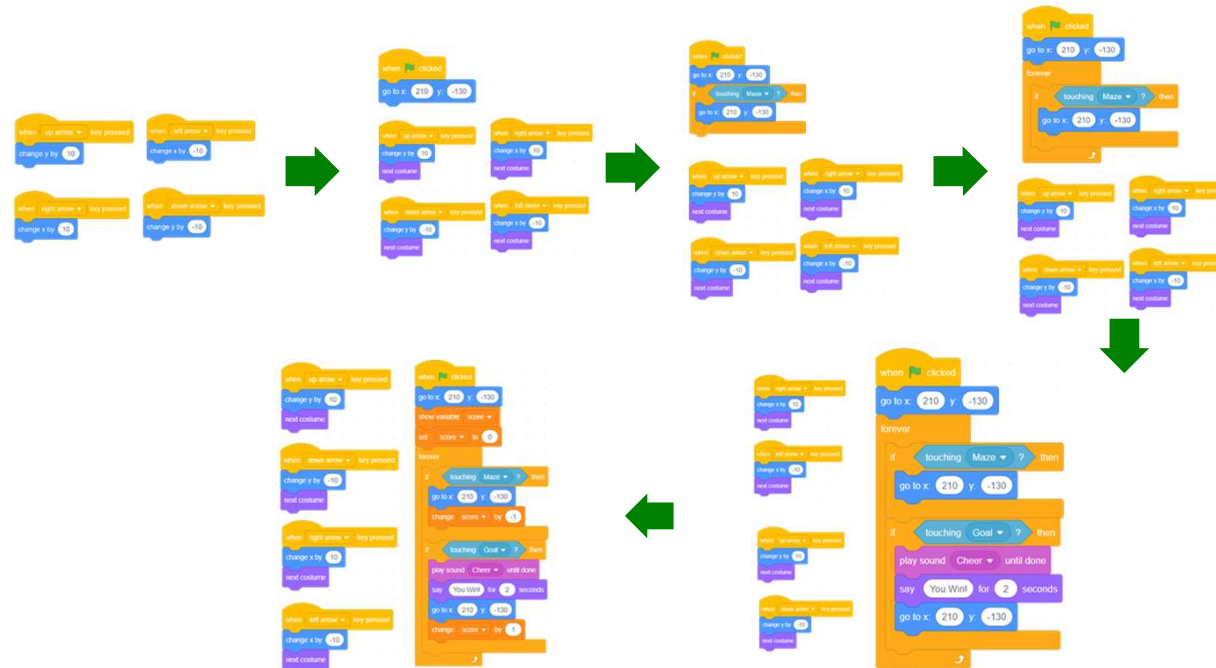
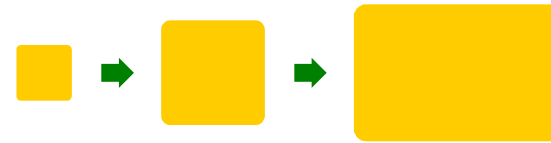
變量 (Variable)



(2) 計算思維主要概念和實踐

反覆構思及漸進編程

Being incremental and iterative:



(2) 計算思維主要概念和實踐

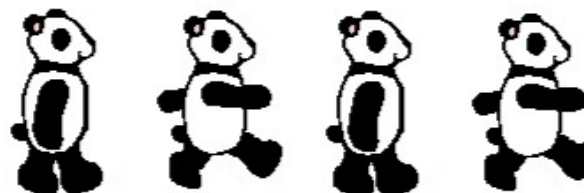
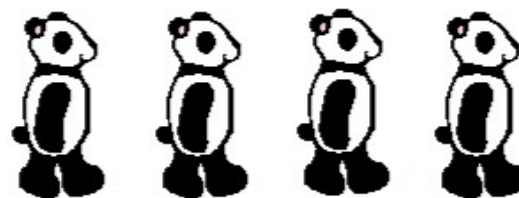
測試及除錯

Testing and debugging:

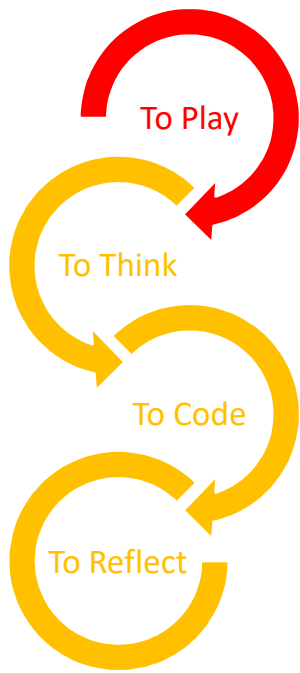
測試



除錯



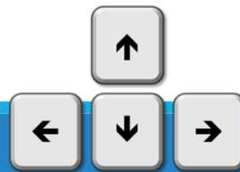
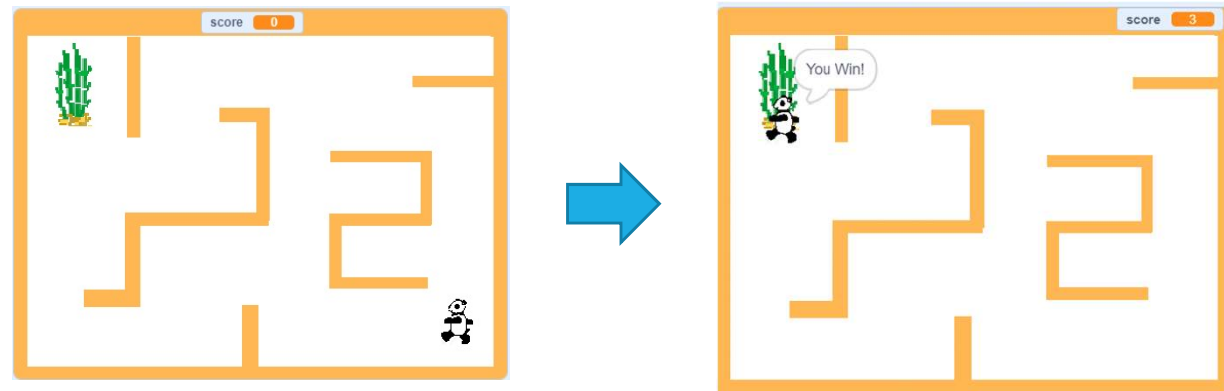
(3) 玩一玩 (To Play)



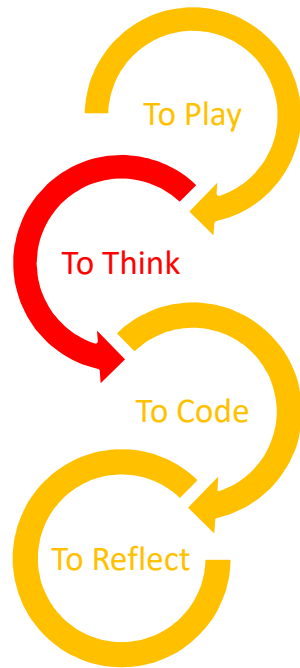
To Play:

- 運用上、下、左、右之方向鍵控制熊貓的方向，前往竹林並避免碰到牆壁。

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



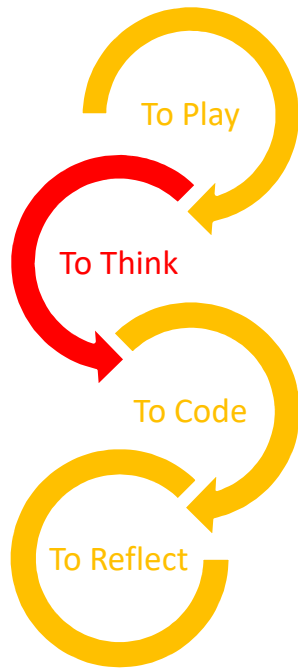
(3) 想一想 (To Think)



To Think:

- 如何使用鍵盤來控制熊貓移動？
- 如何令熊貓看起來像走路？
- 當熊貓碰到牆壁，會發生什麼事情？
- 當熊貓到達竹林，會發生什麼事情？

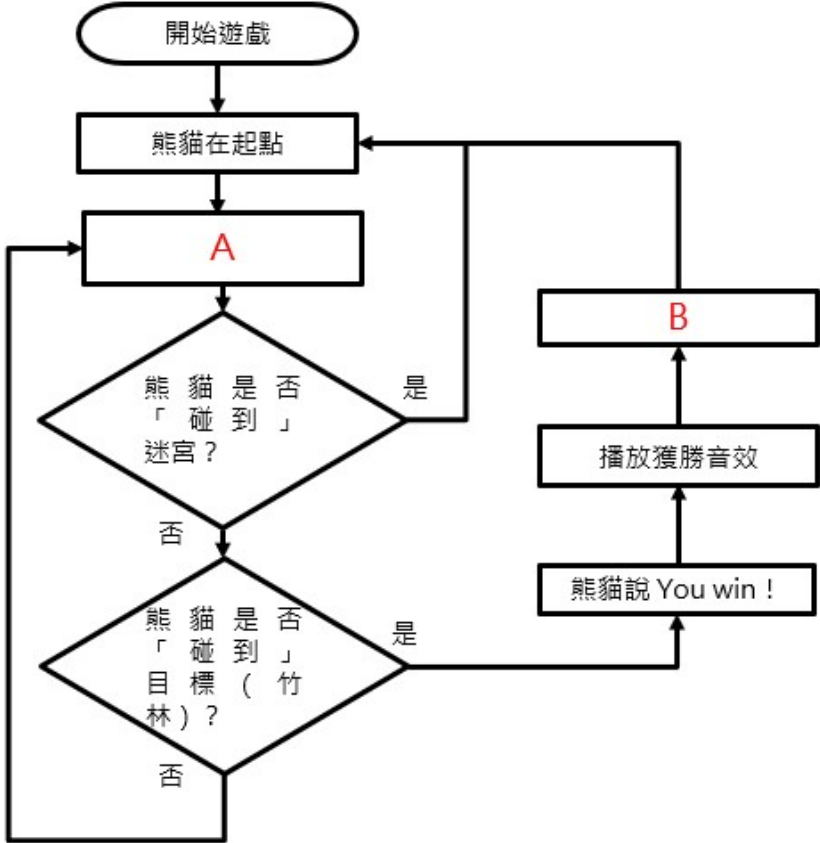
(3) 想一想 (To Think)



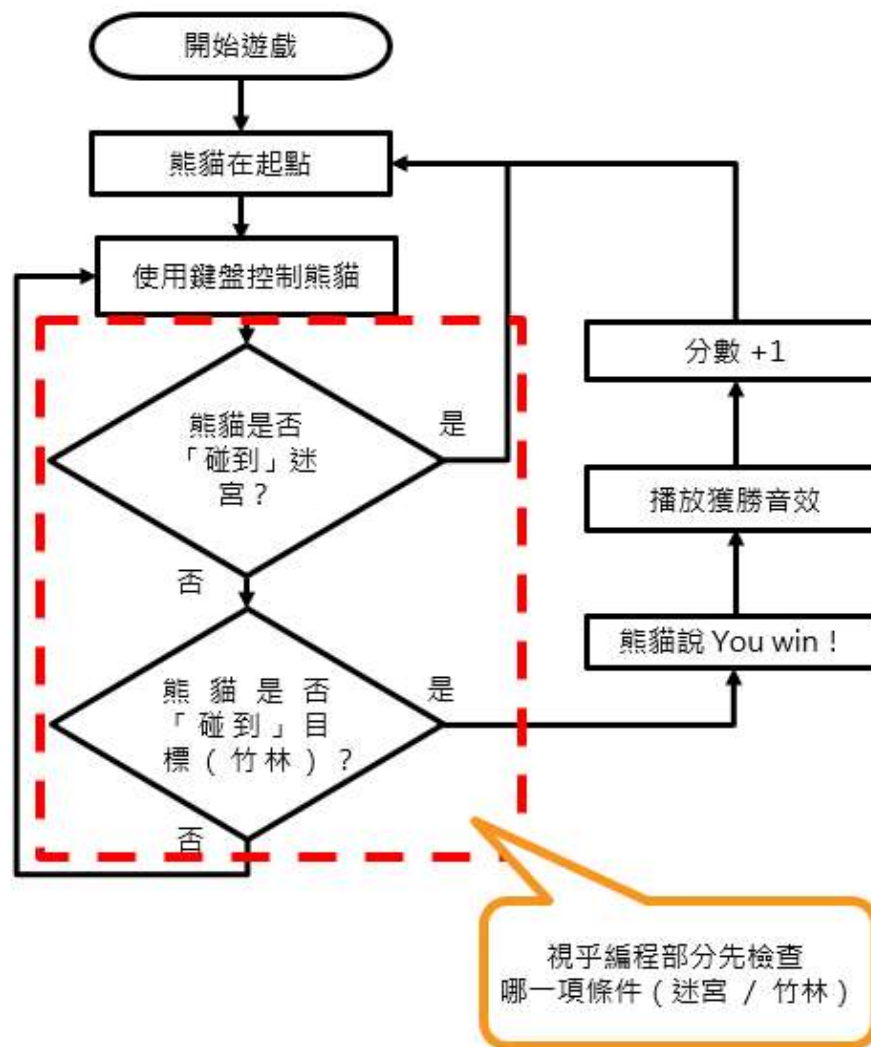
To Think:

- 運用上、下、左、右之方向鍵事件控制熊貓的方向
- 更改熊貓的造型，令熊貓角色看似人一般的走路
- 輸了：如果熊貓「碰到」迷宮，就回到起點
- 勝出：如果熊貓「碰到」竹林，就顯示勝出「You Win！」及發出聲效，分數也會加一分

(3) 流程圖



(3) 流程圖



(3) 工作紙及不插電課堂活動：變量

創作迷宮遊戲 單元五
教學指引：第三教節

不插電活動：變數

教師應該準備一個盒子和一些糖果來運行這個遊戲。

Variables

Make a Variable

candy

教師拿出一個盒子，
在盒子上寫上「糖果」。

set candy to 0 change candy by 1 change candy by -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)

學生可以選擇其中一項，然後教師會做動作。

學生：糖果設為 5，教師從盒子裡取出所有糖果，然後將 5 顆糖果放入盒子裡

學生：糖果改變 -2，教師從盒子裡拿出 2 顆糖果

學生：糖果改變 2，教師在盒子裡放 2 顆糖果

遊戲 2: 猜猜盒子裡有多少顆糖果：

set candy to 10

change candy by 2

12

set candy to 3

change candy by 1

change candy by -1

3

set candy to 6

change candy by 2

change candy by -1

set candy to 0

0

創作迷宮遊戲 單元五
教學指引：第三教節

建構知識：變數

變數用於存儲數值，它具有以下屬性。

- ❑ 變數有名字。

set candy to 0

candy 0

- ❑ 一個變數每次只能存儲一個值。

candy 6

↓

candy 0

↓

candy 10

- ❑ 變數數值可被更新。

candy 6

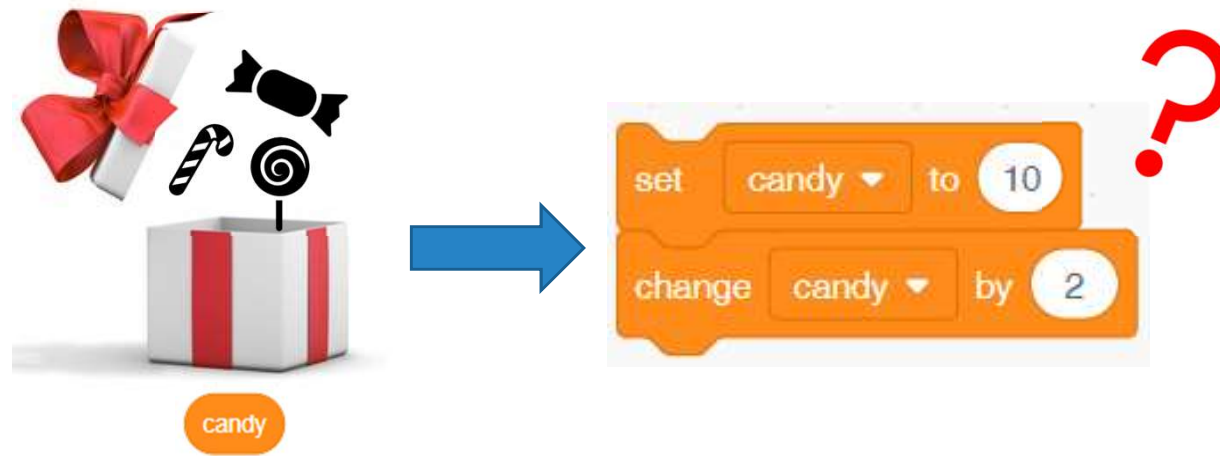
↓

set candy to 0

↑

candy 0

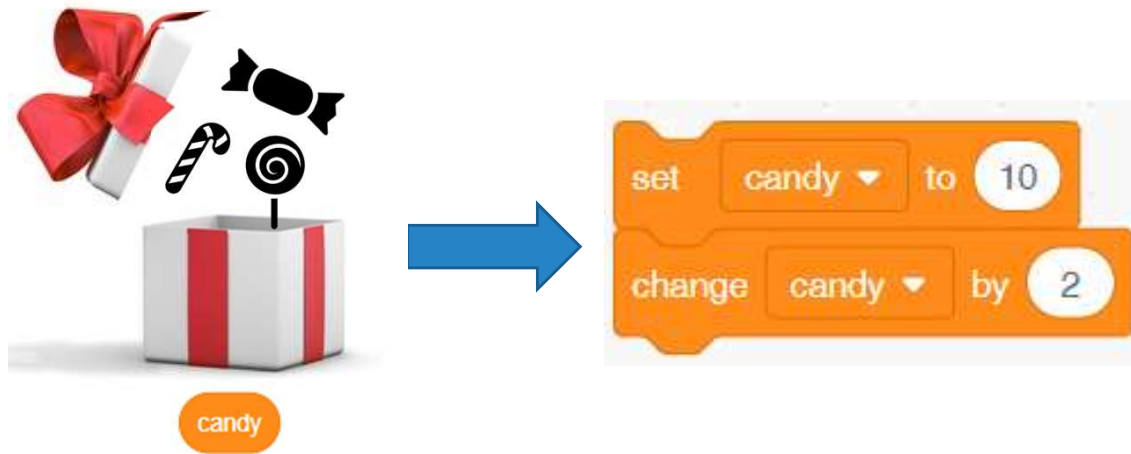
(3) 工作紙及不插電課堂活動：變量



準備一個糖果盒，
名為“candy”，
裏面有一堆糖果

現在盒子裡有多少顆糖果？

(3) 工作紙及不插電課堂活動：變量



準備一個糖果盒，
名為“candy”，
裏面有一堆糖果

現在盒子裡有多少顆糖果？

12

(3) 工作紙及不插電課堂活動：變量



(3) 工作紙及不插電課堂活動：變量



```
set candy to 3
change candy by 1
change candy by -1
```



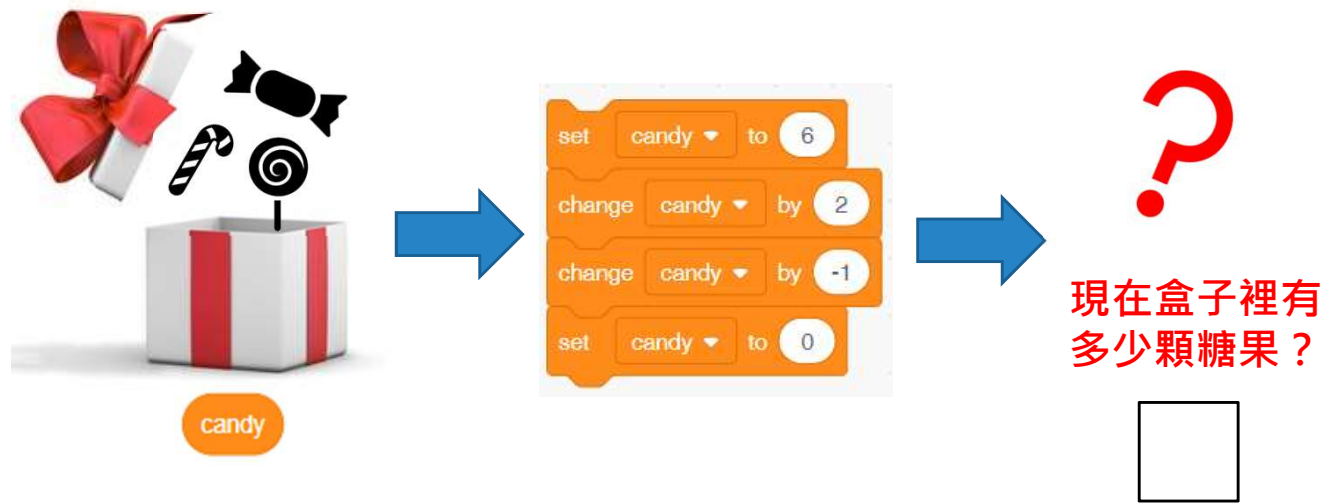
Three orange code blocks stacked vertically. The first block is 'set candy to 3', the second is 'change candy by 1', and the third is 'change candy by -1'.



現在盒子裡有多少顆糖果？

3

(3) 工作紙及不插電課堂活動：變量



(3) 工作紙及不插電課堂活動：變量



```
set candy to 6
change candy by 2
change candy by -1
set candy to 0
```

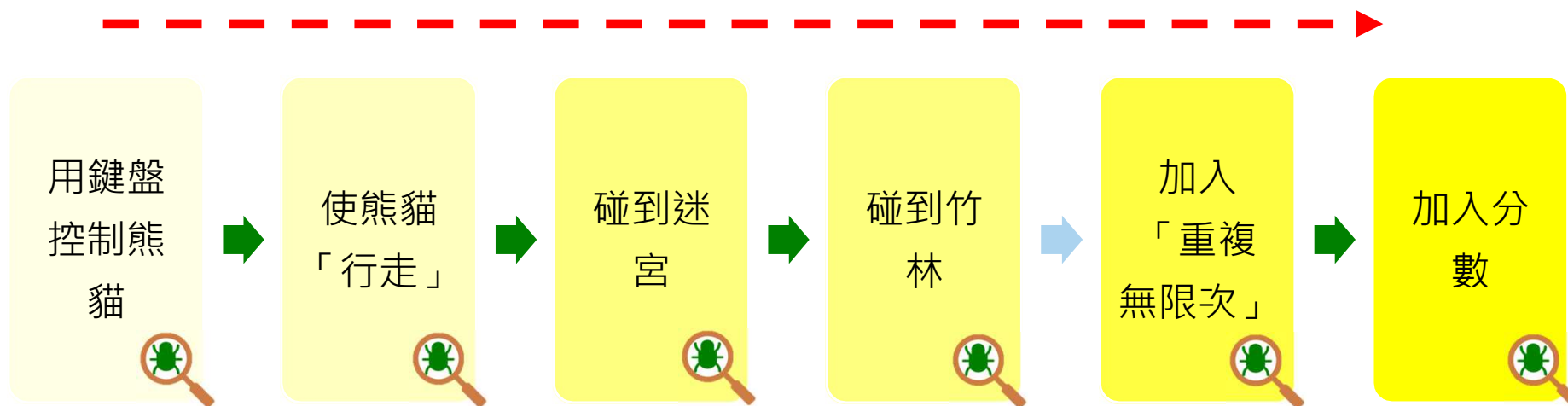


現在盒子裡有多少顆糖果？

0

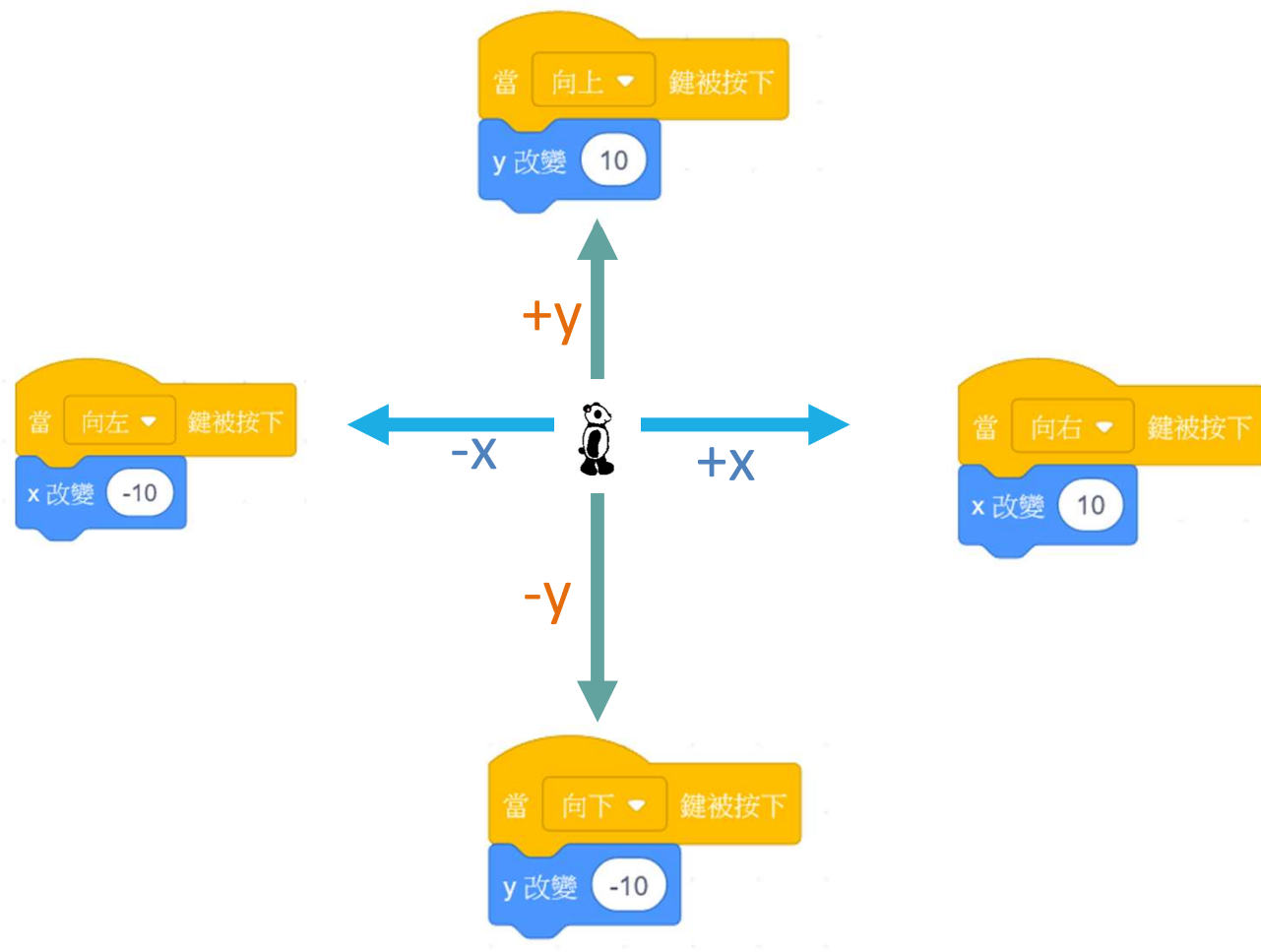
(4) 科技教學學科知識 (TPACK)：製作迷宮遊戲

反覆構思及漸進編程
(Being incremental and iterative)

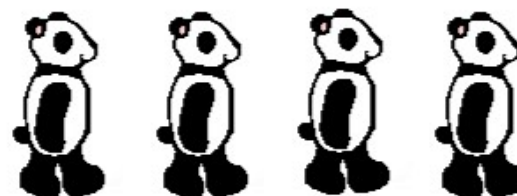


測試及除錯
(Testing and Debugging)

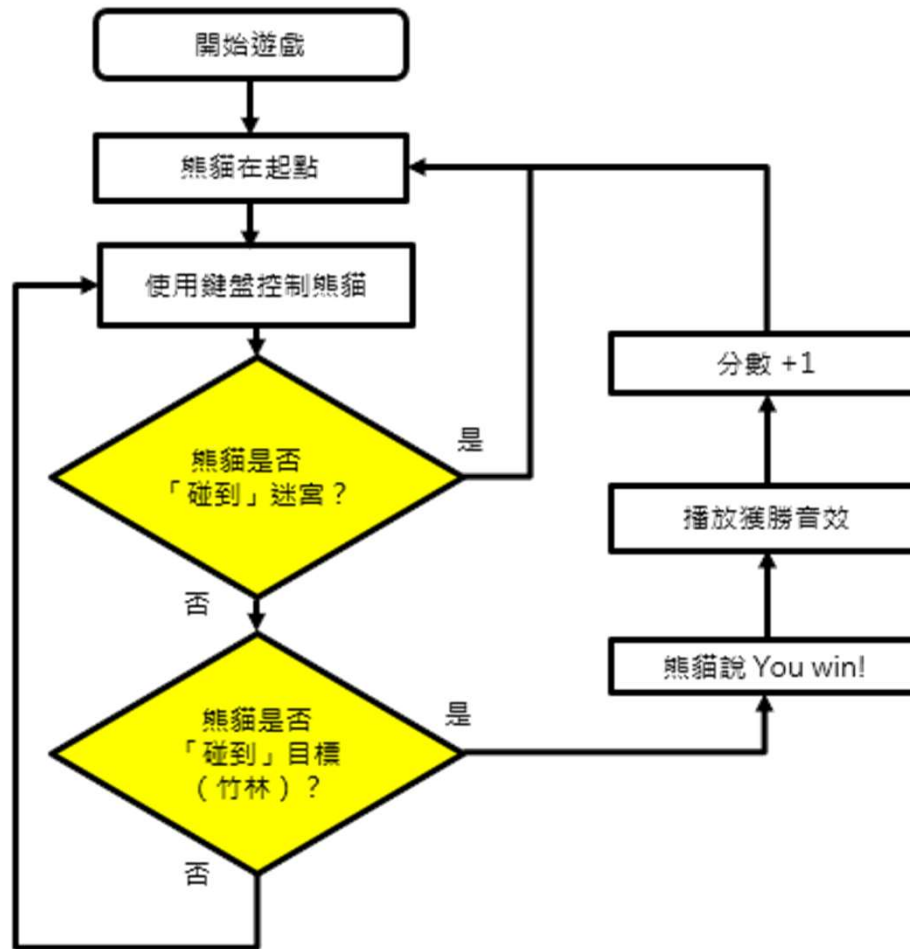
(4) 科技教學學科知識 (TPACK)：用鍵盤控制熊貓



(4) 科技教學學科知識 (TPACK)：使熊貓「行走」



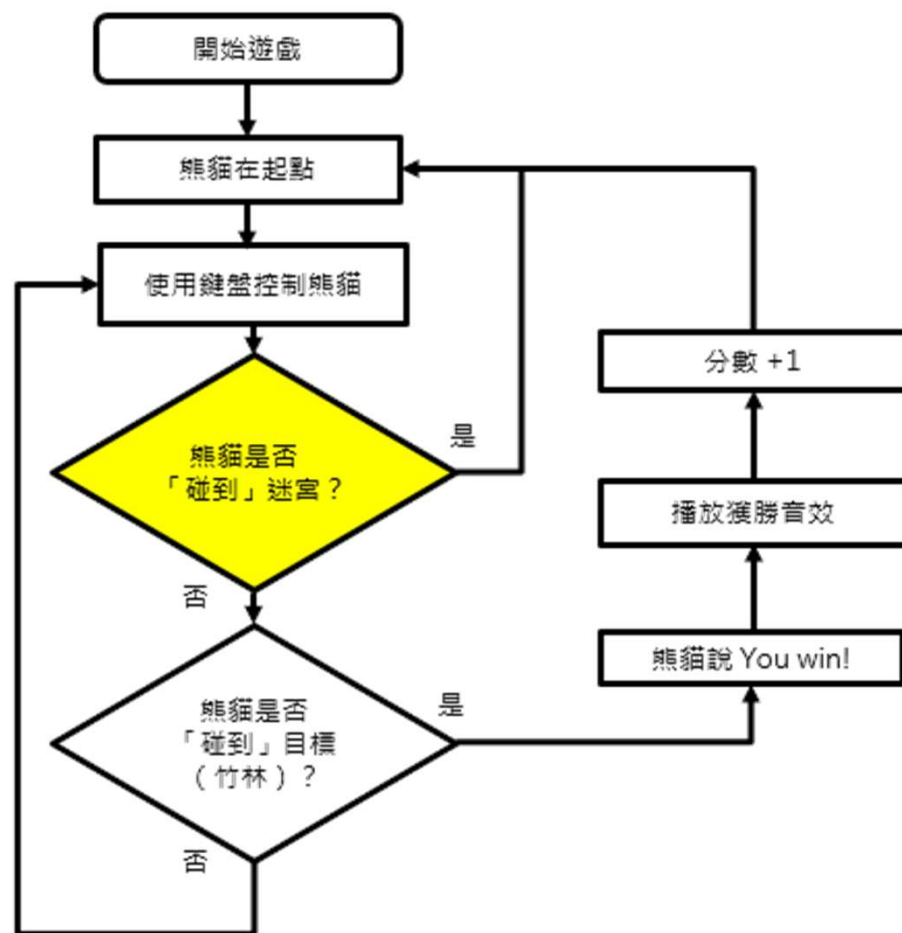
(4) 科技教學學科知識 (TPACK)：碰到迷宮 / 竹林



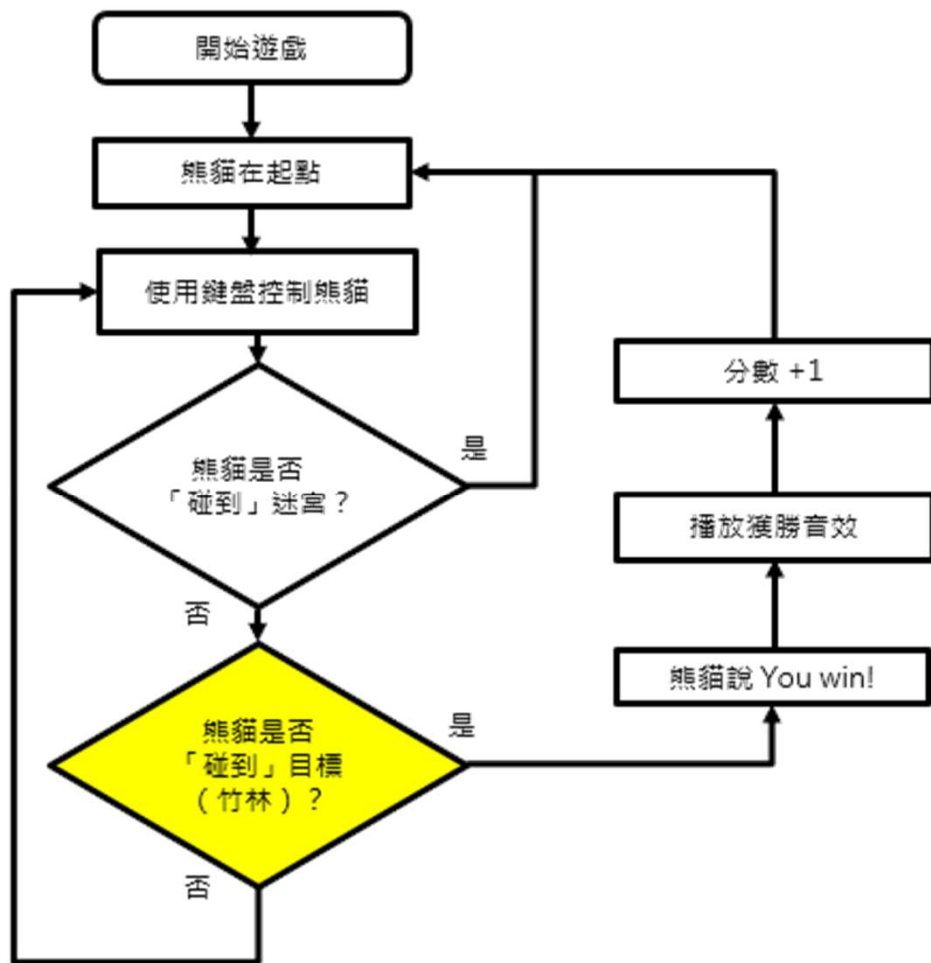
分支/選擇：

熊貓會根據**條件**而作出不同的反應

(4) 科技教學學科知識 (TPACK)：碰到迷宮

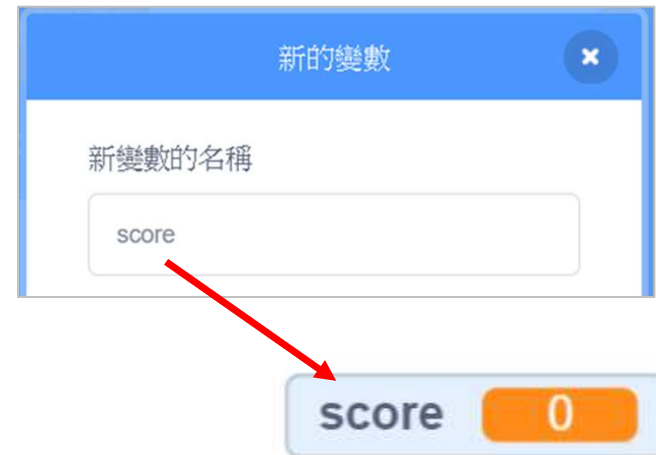


(4) 科技教學學科知識 (TPACK)：碰到竹林



(4) 科技教學學科知識 (TPACK)：加入分數

- 建立及命名變量(Variable)
- 不插電課堂活動：變量(Variable)



(4) 科技教學學科知識 (TPACK)：加入分數

- 初始化變量 (Variable):



- 更新變量:



(5) 科技內容知識反思，啟發數碼創意

提問學生想運用學到的來創造什麼？

✓ 例如：你想創造什麼遊戲？

超市購物遊戲

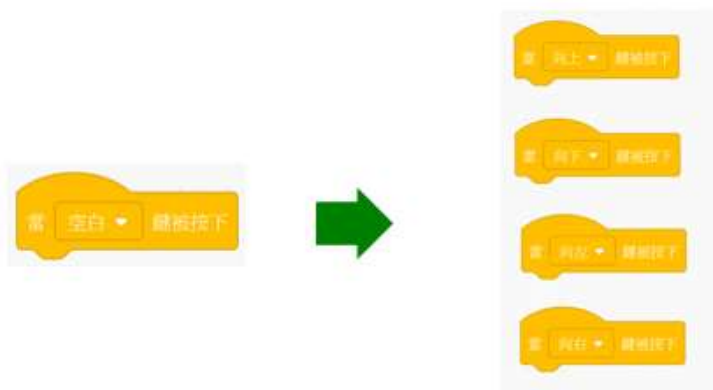
(6) 總結計算思維主要概念和實踐

事件：我們用事件指令方塊來觸發 Scratch 做動作。

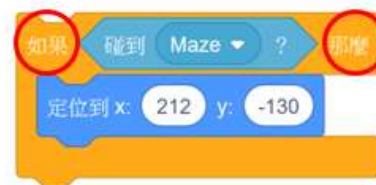


重用及混合程序 / 編碼：

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。



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。

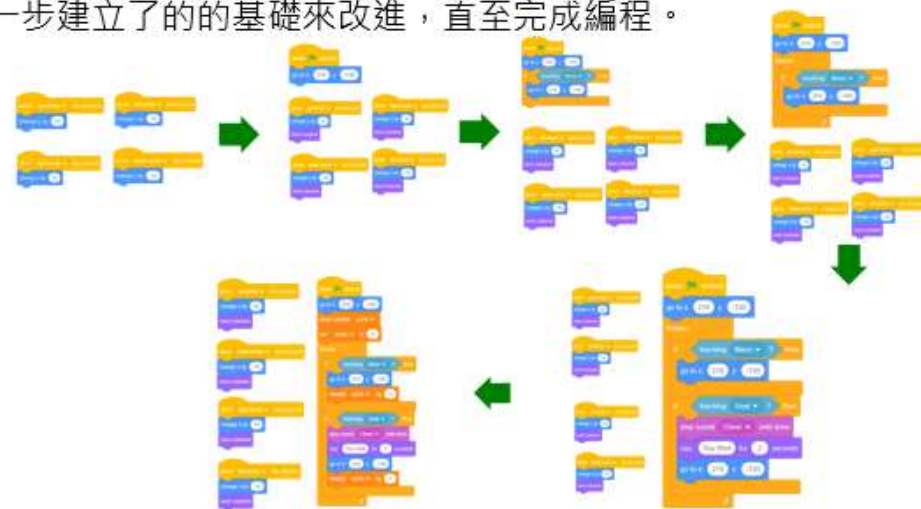


(6) 總結計算思維主要概念和實踐

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。



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。



(6) 總結計算思維主要概念和實踐

事件：

- 除了點擊 ，可以如何觸發動作？



The image shows four Scratch event blocks, each enclosed in a red rounded rectangle. Each block consists of a yellow event block, a blue motion block, and a purple 'change sprite' block.

- Top-left: Event: '當 向上 鍵被按下' (When Up key is pressed). Motion: 'y 改變 10' (y change 10). Action: '造型換成下一個' (Change sprite to next).
- Top-right: Event: '當 向下 鍵被按下' (When Down key is pressed). Motion: 'y 改變 -10' (y change -10). Action: '造型換成下一個' (Change sprite to next).
- Bottom-left: Event: '當 向左 鍵被按下' (When Left key is pressed). Motion: 'x 改變 -10' (x change -10). Action: '造型換成下一個' (Change sprite to next).
- Bottom-right: Event: '當 向右 鍵被按下' (When Right key is pressed). Motion: 'x 改變 10' (x change 10). Action: '造型換成下一個' (Change sprite to next).

答案：
按下鍵盤。

(6) 總結計算思維主要概念和實踐

分支 / 選擇:

- 如果熊貓碰到迷宮，會發生什麼事？

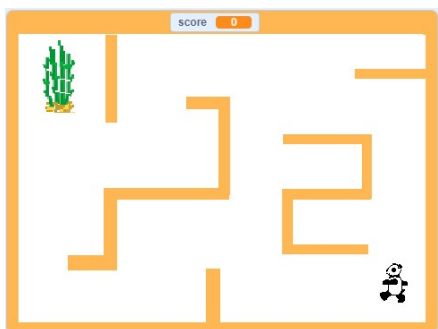


答案: 返回起點。

(6) 總結計算思維主要概念和實踐

循環 – 重複無限次:

- 倘若你不加入「重複無限次」指令方塊？



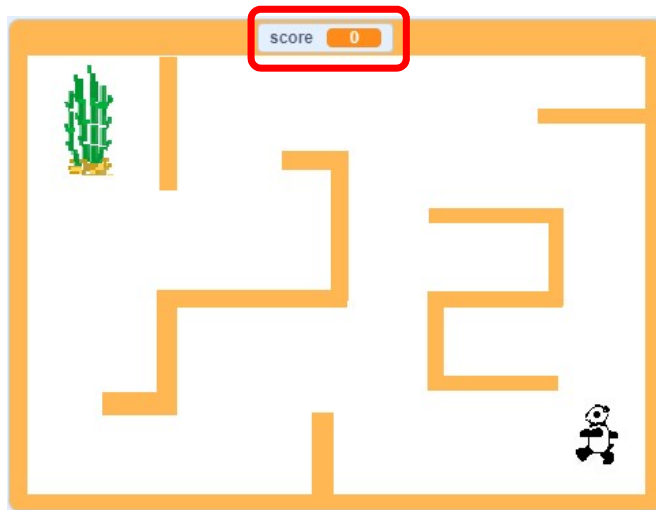
答案：
條件只會執行一次。「重複無限次」指令方塊能讓程序不斷檢查有否出現碰撞。



(6) 總結計算思維主要概念和實踐

變量:

- 當你勝出時，分數會有什麼變更？



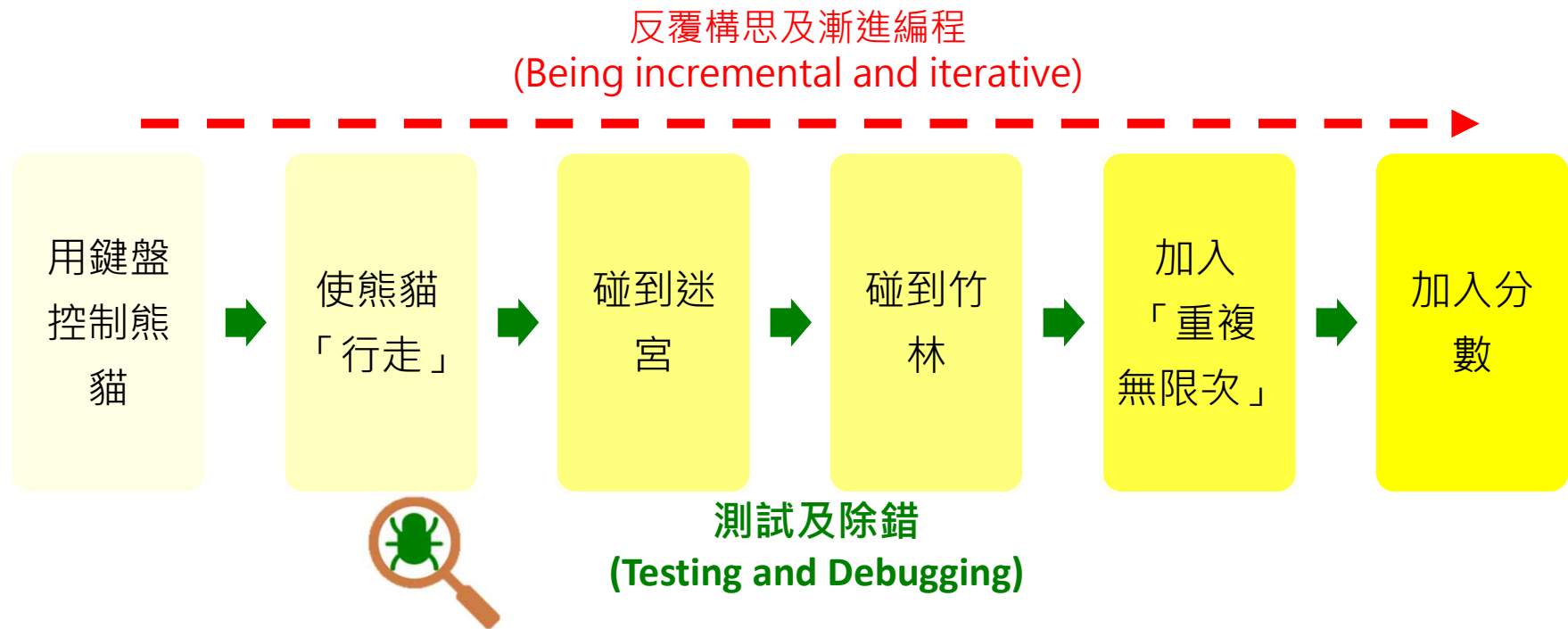
Answer: Score + 1.



(6) 總結計算思維主要概念和實踐

測試及除錯:

你如何確保這些功能能夠運作？



(7) 教師自我反思教學法

- 於本教節，你運用了什麼教學法？
- 課堂活動及工作紙有用嗎？
- 你有沒有其他方法教授變量 (variable)？



Final Project

專題習作



為什麼要做專題習作？

- 展示計算思維概念、實踐、態度
- 培養解難能力
- 激發數碼創意

學習目標

1. 學生能定義一個待解決的問題，並提出建議以運用Scratch編程解決該問題；
2. 設計一個Scratch專案來解決定義的問題；
3. 應用計算思維概念、實踐和態度來完成 Scratch 專案；和
4. 通過創造自己的編程作品，培養學生的創造力；並通過與同學和教師分享自己的專案，展示編程可以成為一種有趣的社交活動。

重溫計算思維主要概念和實踐

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！



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。



變量

變量可以幫助你在 Scratch 專案中儲存和更新數值。



命名

給角色起個合適名字也很重要。



重溫計算思維主要概念和實踐

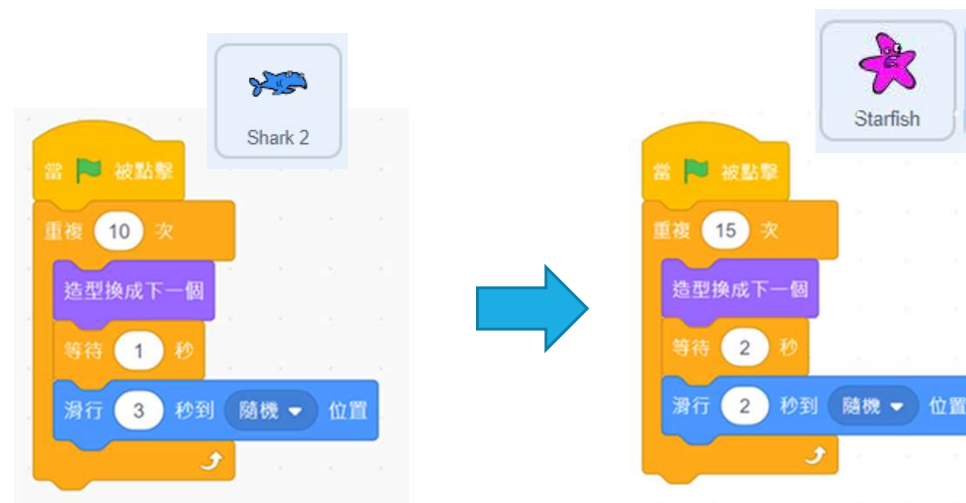
函式積木

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。



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。



重溫基本程序編寫結構的概念

序列

程序的序列是指執行編程指令的次序。



分支 / 選擇

可以透過「如果 – 那麼」指令方塊來決定角色的動作。



循環

「重複無限次」和「重複」可引發重覆的動作。



重溫基本程序編寫結構的概念

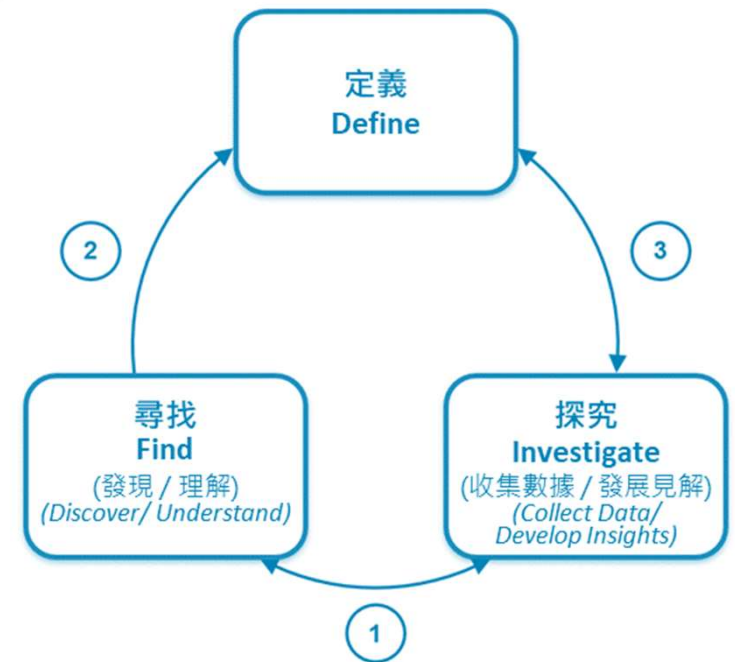
條件邏輯運算符

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。



引導學生定義問題、尋找問題解決方案

- **Guide students to identify a problem**
引導學生定義問題
- **Analyze the problem**
分析問題
- **Look for solutions by designing the algorithm and programs**
尋找問題解決方案並設計算法及程序
- **Make reference to the Scratch units**
重溫並參考所學單元知識



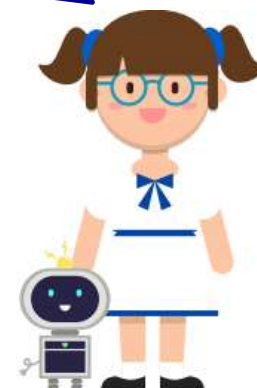
(Kong, 2022)

同創作 (To create)

同創作 (To create)：透過修改專案並加入自己的構思，學生對跟他人分享自己製作的專案而感到自豪。



我想寫一個遊戲俾
我朋友仔玩！



我想為我的嫲嫲創
作一個動畫故事。

定義問題

1. 你身邊有沒有人遇到一些問題？她 / 他是誰？

我的祖母。

2. 他 / 她面臨什麼問題？

祖母想了解科技的發展，但較少機會接觸新科技。

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。

故事

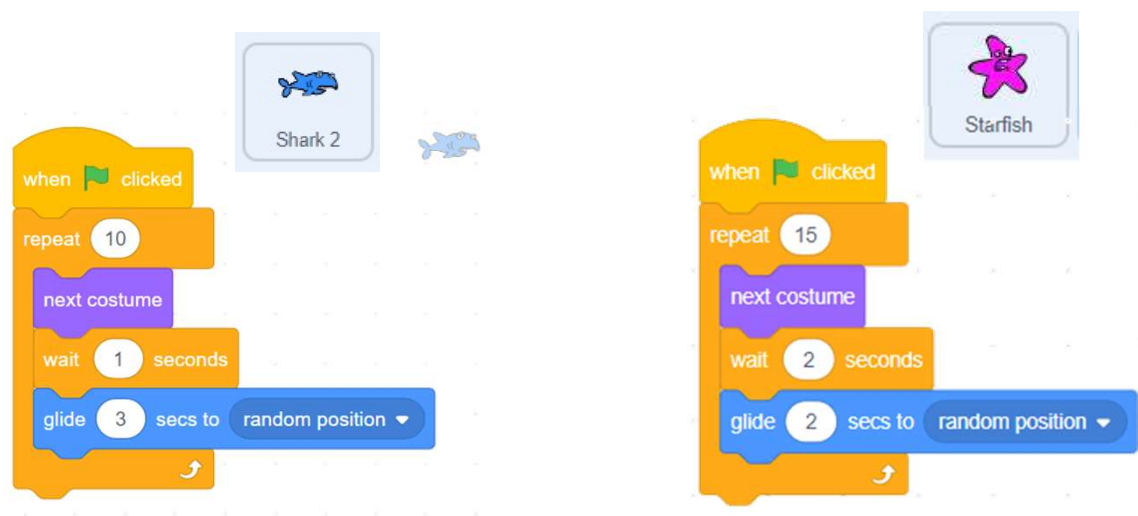
我可以用 Scratch 設計一個故事動畫，為祖母寫一個故事。在故事中，我和爸爸會帶著祖母在元宇宙中環遊世界。

我想為我的麻麻創作一個動畫故事。



重用及混合已學單元的程序 / 編碼

我們可以重用及混合已學單元的程序 / 編碼，應用到專題研習中。



啓發學生數碼創意

為了更好地激發學生的數碼創意，教師可以：

- 鼓勵對相同主題有興趣的學生分為一組
- 將相似題目的小組配對，進行互評
- 給學生適切的指引，例如，舉出例子引發學生思考日常生活中遇到的問題

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？



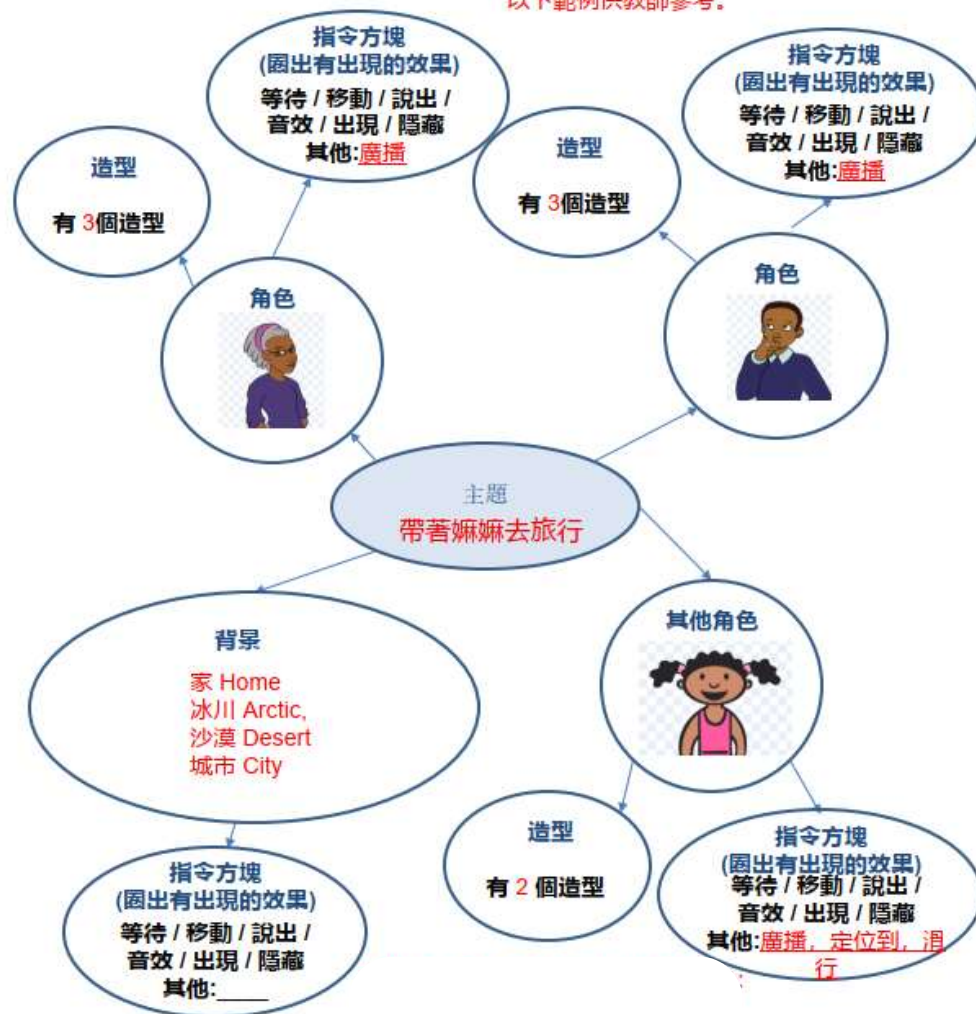
你的朋友喜歡音樂！你能製作一個可以播放不同音樂的音樂盒嗎？



示例一 詳細定義問題

利用腦圖列出故事中的元件。

以下範例供教師參考。



範例“帶著嫻嫻去旅行”供教師參考：

<https://scratch.mit.edu/projects/777900161/>

示例一 提出解決問題方案（優異作品）

利用分鏡劇本，按序繪製圖像來設計專案。

範例（1）「帶著祖母去旅行」（優異）
供教師參考：

<https://scratch.mit.edu/projects/777900161>



示例一 提出解決問題方案（良好作品）

利用分鏡劇本，按序繪製圖像來設計專案。

範例（1）「帶著祖母去旅行」（良好）
供教師參考：

<https://scratch.mit.edu/projects/822750642>



示例二 提出解決問題的設計（優異作品）

設計遊戲

範例（2）「水果遊戲」（優異）供教師參考：
<https://scratch.mit.edu/projects/845208981/>



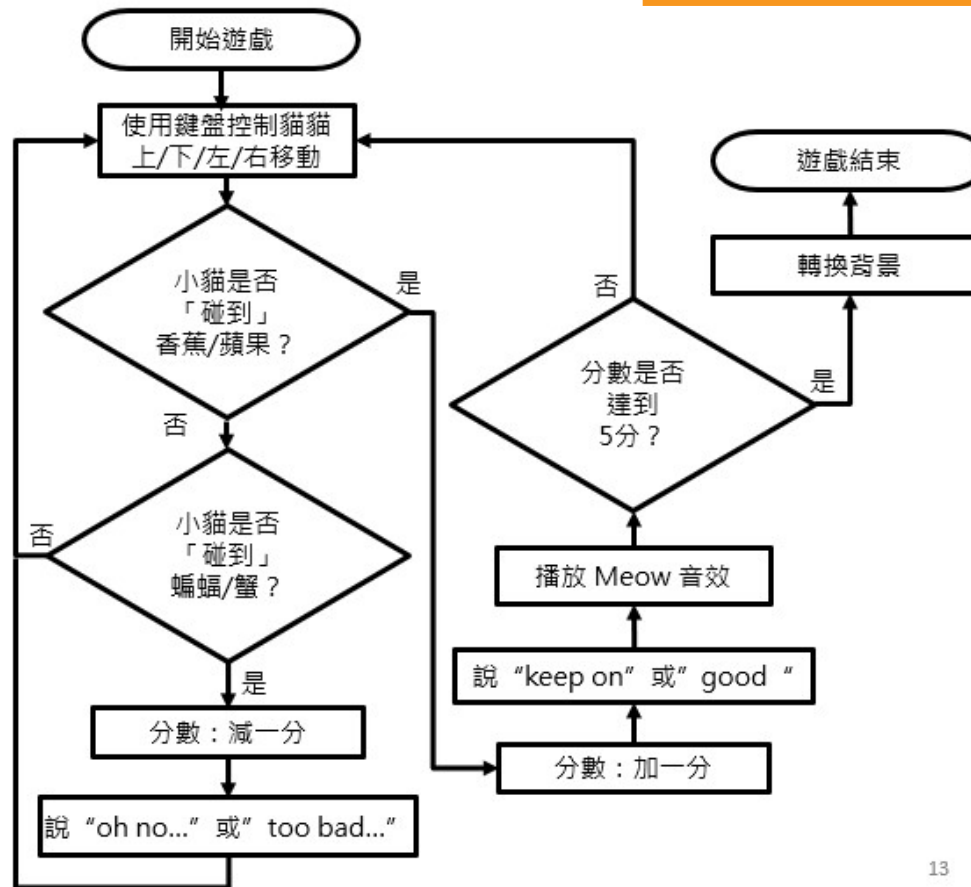
遊戲設計

角色	蘋果、香蕉、蝙蝠、蟹、貓貓
動作	上/下/左/右鍵、碰到、分數
遊戲設計 / 規則	<p>使用鍵盤（上/下/左/右鍵）控制小貓吃生果。</p> <ul style="list-style-type: none">➢ 生果和障礙物會隨機出現。➢ 每吃一個生果加一分，並會發出Meow音效。➢ 每次碰到障礙物扣一分。➢ 拿到五分會勝出遊戲。

示例二 提出解決問題的設計 (優異作品)

利用流程圖設計遊戲

範例 (2) 「水果遊戲」 (優異) 的流程圖供教師參考



示例二 提出解決問題的設計（良好作品）

設計遊戲

範例（2）「水果遊戲」（良好）供教師參考：
<https://scratch.mit.edu/projects/845198792/>



遊戲設計

角色	貓貓、蘋果
動作	上/下/左/右鍵、碰到、分數
遊戲設計 / 規則	<p>使用鍵盤（上/下/左/右鍵）控制小貓吃生果。</p> <ul style="list-style-type: none">➢ 蘋果會隨機出現。➢ 每吃一個生果加一分。➢ 拿到十分會勝出遊戲。

示例二 提出解決問題的設計（尚可作品）

設計遊戲

範例（2）「配對遊戲」（尚可）供教師參考：
<https://scratch.mit.edu/projects/845174154/>




遊戲設計

角色	小貓、蘋果
動作	上/下/左/右鍵、碰到
遊戲設計/ 規則	使用鍵盤（上/下/左/右鍵）控制小貓吃生果。 <ul style="list-style-type: none">➤ 蘋果會在特定地方隨時出現。➤ 小貓吃掉蘋果後，蘋果會消失。

與另一小組同儕互評

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：「我很喜歡你們程式的背景。」 「角色生動有趣。」



「我希望角色小貓可以在不同的場景中轉換不同的造型。」



「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



全班小組簡報互評

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**程式的背景**。」  「**角色**生動有趣。」

 「我希望**角色**小貓可以在不同的場景中轉換不同的造型。」  「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



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..... _____

專題習作教節編排

To-Do Checklist		
日期/教節	要完成的任務	狀態 (已完成 / 遇到問題 , 解決方案)
例 , 第一教節	設計流程圖	於第一教節完成
第二教節	在場景1中添加角色和背景	於第二教節完成

專題習作評分標準 (小組)

範疇	優異	良好	尚可	繼續努力
為專題習作定義問題 (16%)	能透過仔細的觀察和具體的分析來清楚地辨別和定義專題習作需解決的問題，從而提出相關可行的解決方案/設計，並制定周詳計劃來解決該問題。 分數：13 - 16	能透過觀察和分析來辨別和定義專題習作需解決的問題，且提出相關可行的解決方案/設計，並制定合適的計劃來解決問題。 分數：9 - 12	基本能辨別和定義專題習作需解決的問題，且提出相關可行的解決方案/設計，並制定簡單計劃來解決問題。 分數: 5 - 8	未能明確辨別和定義專題習作需解決的問題，提出的解決方案/設計不能用於解決該問題。 分數: 0 - 4
運用計算思維概念和實踐進行算法設計 (40%)	程序設計中呈現合乎邏輯的序列(sequence)。程序以甚為有效的方式設計，能出色地使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，從而進一步簡化和優化程序，高效地完成所有任務。 分數: 31 - 40	程序設計中呈現合乎邏輯的序列(sequence)。程序以有效的方式設計，能良好地使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，從而簡化和優化程序，充分完成任務。 分數: 21 - 30	程序設計中頗能呈現合乎邏輯的序列(sequence)，大致能恰當使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，完成任務，表現尚算令人滿意。 分數: 11 - 20	程序設計中缺乏合乎邏輯的序列(sequence)，程序中未能嘗試使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，任務未能完成。 分數: 0 - 10
應用Scratch 編程環境中的元件和指令方塊 (20%)	在需要解決問題的情境中，能出色地利用Scratch 編程環境中的元件和指令方塊，創建出互動和讓用戶積極參與的程序。 分數: 16 - 20	在需要解決問題的情境中，能善用Scratch 編程環境中的元件和指令方塊，創建出能為用戶帶來具參與性的程序。 分數: 11 - 15	在需要解決問題的情境中，能恰當地使用Scratch元件和指令方塊創建出合適的程序。 分數: 6 - 10	在需要解決問題的情境中，只能有限地運用Scratch元件和指令方塊創建程序，未能達致預期效果。 分數: 0 - 5

專題習作評分標準 (小組)

<p>美觀 (12%)</p>	<p>專題習作設計別出心裁，充份發揮創意潛能，具有高水平的視覺和感官吸引力。注重細節充分地運用感官，成功增強用戶體驗。</p> <p>分數: 10 - 12</p>	<p>專題習作展示了一定的創意和良好的視覺設計。注重細節，善用聲音、視覺等感官元素來吸引用戶。</p> <p>分數: 5 - 9</p>	<p>專題習作嘗試展示創意，具合適的視覺設計。恰當地運用聲音、視覺等感官元素來吸引用戶。</p> <p>分數: 4 - 6</p>	<p>專題習作未能展示創意，欠缺視覺設計。未能恰當地運用聲音、視覺等感官元素吸引用戶。</p> <p>分數: 0 - 3</p>
<p>與同儕合作 (12%)</p>	<p>團隊成員之間的溝通和協作非常有效，有強烈的團隊精神。團隊成員能相互合作，有效地完成任務。每位團隊成員都有明確的角色，共同協作實現共同的目標，彼此合作中表現出高度的信任和尊重。</p> <p>分數: 10 - 12</p>	<p>團隊成員之間的溝通和協作有效，有良好的團隊精神。團隊成員相互合作，完成任務。每位團隊成員都有不同的角色，共同協作實現共同的目標，彼此合作中表現出信任和尊重。</p> <p>分數: 7 - 9</p>	<p>團隊成員能溝通和協作，有團隊精神。每位團隊成員都有不同的角色，大部分團隊成員能透過彼此合作，完成任務，實現共同的目標。</p> <p>分數: 4 - 6</p>	<p>團隊成員的溝通和協作有限，缺乏團隊精神。大多數團隊成員獨立工作，未能互相幫助解決問題，致使任務未能完成。</p> <p>分數: 0 - 3</p>
<p>總分100</p> <p>優異：76 – 100 良好：51 – 75 尚可：26 – 50 繼續努力：0 – 25</p>				

專題習作評分標準 (個人)

範疇	優異	良好	尚可	繼續努力
為專題習作定義問題 (20%)	能透過仔細的觀察和具體的分析來清楚地辨別和定義專題習作需解決的問題，從而提出相關可行的解決方案/設計，並制定周詳計劃來解決該問題。 分數: 16 - 20	能透過觀察和分析來辨別和定義專題習作需解決的問題，且提出相關可行的解決方案/設計，並制定合適的計劃來解決問題。 分數: 11 - 15	基本能辨別和定義專題習作需解決的問題，且提出相關可行的解決方案/設計，並制定簡單計劃來解決問題。 分數: 6 - 10	未能明確辨別和定義專題習作需解決的問題，提出的解決方案/設計不能用於解決該問題。 分數: 0 - 5
運用計算思維概念和實踐進行算法設計 (40%)	程序設計中呈現合乎邏輯的序列(sequence)。程序以甚為有效的方式設計，能出色地使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，從而進一步簡化和優化程序，高效地完成所有任務。 分數: 31 - 40	程序設計中呈現合乎邏輯的序列(sequence)。程序以有效的方式設計，能良好地使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，從而簡化和優化程序，充分完成任務。 分數: 21 - 30	程序設計中頗能呈現合乎邏輯的序列(sequence)，大致能恰當使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，完成任務，表現尚算令人滿意。 分數: 11 - 20	程序設計中缺乏合乎邏輯的序列，程序中未能嘗試使用循環(iteration)、分支/選擇(branching / selection)和變量(variables)等，任務未能完成。 分數: 0 - 10

專題習作評分標準 (個人)

<p>應用Scratch 編程環境中的元件和指令方塊 (20%)</p>	<p>在需要解決問題的情境中，能出色地利用Scratch 編程環境中的元件和指令方塊，創建出互動和讓用戶積極參與的程序。 分數: 16 - 20</p>	<p>在需要解決問題的情境中，能善用Scratch 編程環境中的元件和指令方塊，創建出能為用戶帶來具參與性的程序。 分數: 11 - 15</p>	<p>在需要解決問題的情境中，能恰當地使用Scratch元件和指令方塊，創建出合適的程序。 分數: 6 - 10</p>	<p>在需要解決問題的情境中，只能有限地運用Scratch元件和指令方塊創建程序，未能達致預期效果。 分數: 0 - 5</p>
<p>美觀 (20%)</p>	<p>專題習作設計別出心裁，充份發揮創意潛能，具有高水平的視覺和感官吸引力。注重細節，充分地運用感官，成功增強用戶體驗。 分數: 16 - 20</p>	<p>專題習作展示了一定的創意和良好的視覺設計。注重細節，善用聲音、視覺等感官元素來吸引用戶。 分數: 11 - 15</p>	<p>專題習作嘗試展示創意，具合適的視覺設計。恰當地運用聲音、視覺等感官元素來吸引用戶。 分數: 6 - 10</p>	<p>專題習作未能展示創意，欠缺視覺設計。未能恰當地運用聲音、視覺等感官元素吸引用戶。 分數: 0 - 5</p>

總分100

優異：76 – 100

良好：51 – 75

尚可：26 – 50

繼續努力：0 – 25

References

EDB (2020). *Computational Thinking – Coding Education: Supplement to the Primary Curriculum*

Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)?. *Contemporary issues in technology and teacher education*, 9(1), 60-70.

Kong, S. -C. (2022). Problem formulation in computational thinking development for nurturing creative problem solvers in primary School. *Education and Information Technologies*, 1-20. <https://doi.org/10.1007/s10639-022-11101-9>

Kong, S. -C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127, 178–189. <https://doi.org/10.1016/j.compedu.2018.08.026>

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. <https://doi.org/10.1016/j.compedu.2020.103872>

References

- Kong, S. -C., & Wang, Y. Q. (2020). Formation of computational identity through computational thinking perspectives development in programming learning: A mediation analysis among primary school students. *Computers in Human Behavior*, 106, 106230. <https://doi.org/10.1016/j.chb.2019.106230>
- Kong, S. -C., Lai, M., & Li, Y.G. (2022). Scaling up a teacher development programme for sustainable education in computational thinking: Studies with TPACK surveys, concept tests and primary school visits. *Computers and Education*, 194, 104707. <https://doi.org/10.1016/j.compedu.2022.104707>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Enriched Module on Coding Education for Upper Primary Level – The Design and Pedagogy

高小增潤編程教育課程單元-課程單元 設計及教學法

香港教育大學 江紹祥教授

Professor Kong Siu-Cheung,

Research Chair Professor of E-Learning and Digital Competency

The Education University of Hong Kong

16 June 2023 二零二三年六月十六日

