Briefing Session on the "Enriched Module on Coding Education for Upper Primary Level – Primary 6" (New)

STEAM教育學與教和評估系列：「高小增潤編程教育課程單元 – 小六」簡介會(新辦)

香港教育大學 江紹祥教授
Professor Kong Siu-Cheung,
The Education University of Hong Kong

11 July 2024 二零二四年七月十一日

請掃描二維碼，填寫出席記錄

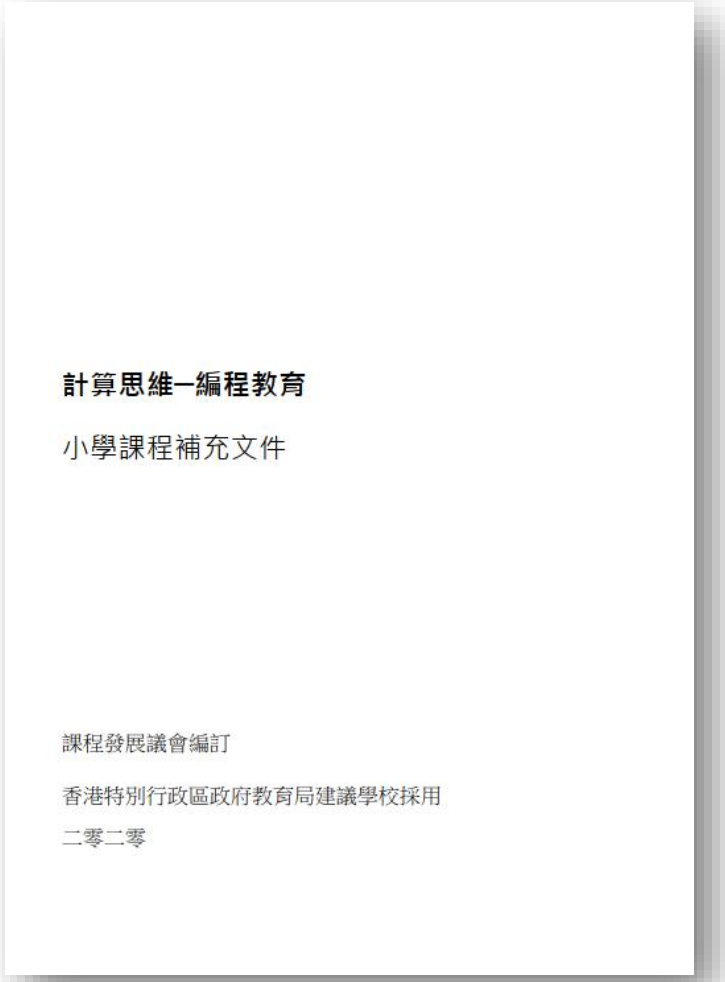
https://bit.ly/3V8uHob

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。

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。



參考資料：
https://www.edb.gov.hk/attachment/tc/curriculum-development/renewal/CT/supplement_CT_Chi_2020.pdf

計算思維 - 編程教育及高小學與教資源設計 – 簡介會目標：

(1) 於小學推動創新科技教育；

(2) 介紹「高小增潤編程教育課程單元 – 小六」，讓學校更有系統地全面落實在高小各級推行編程教育；及

(3) 支援教師於高小推行增潤編程教育。

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？

Mentimeter

https://www.menti.com/alkr9rmpmq1c

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

高小增潤編程教育課程單元（小六）

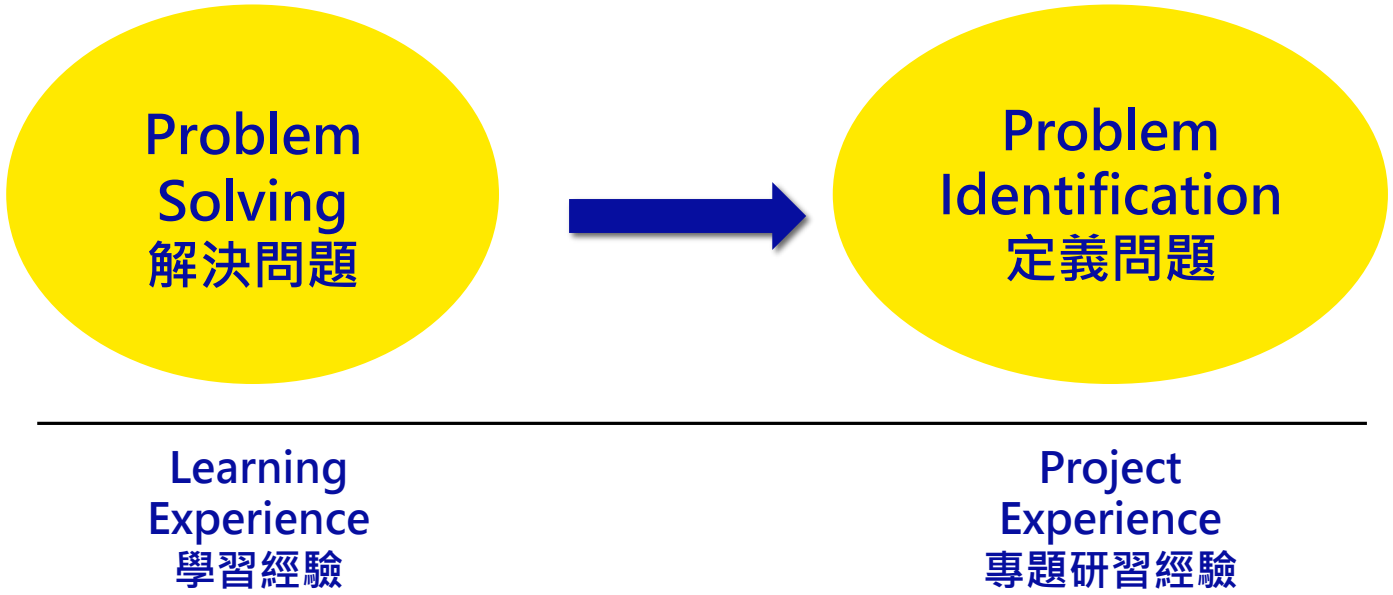
To be confirmed based on Preface

本小學六年級的課程單元：

- 以學生在小四和小五學到的計算思維概念和技能為基礎，加深他們對抽象化、算法和自動化的認識；
- 通過編程及連接實物、運用感測器和執行器與環境進行互動等解決日常問題；
- 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；
- 小六課程還將運算思維與各學科結合。範例包括：
  單元二：數碼鋼琴（音樂科）
  單元五：加法遊戲（數學科）
  單元六：普通話聽力測驗示範（中文科）
  單元七：香港旅遊熱點（常識／人文科）

參考資料：
https://www.edb.gov.hk/tc/curriculum-development/kla/technology-edu/resources/InnovationAndTechnologyEducation/resources.html

Nurture Students' Problem-solving Skills and Digital Creativity
培養學生的解難能力和數碼創意

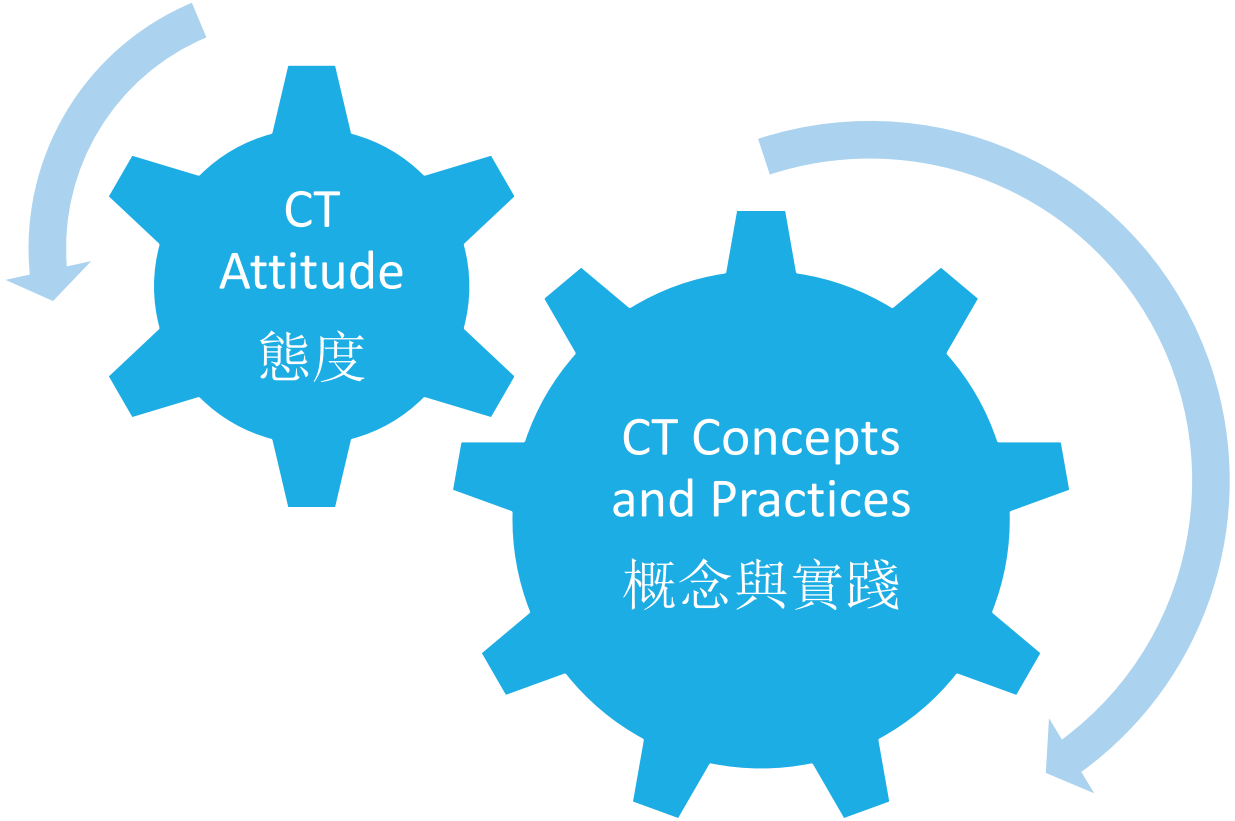


Computational Thinking 計算思維

- 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

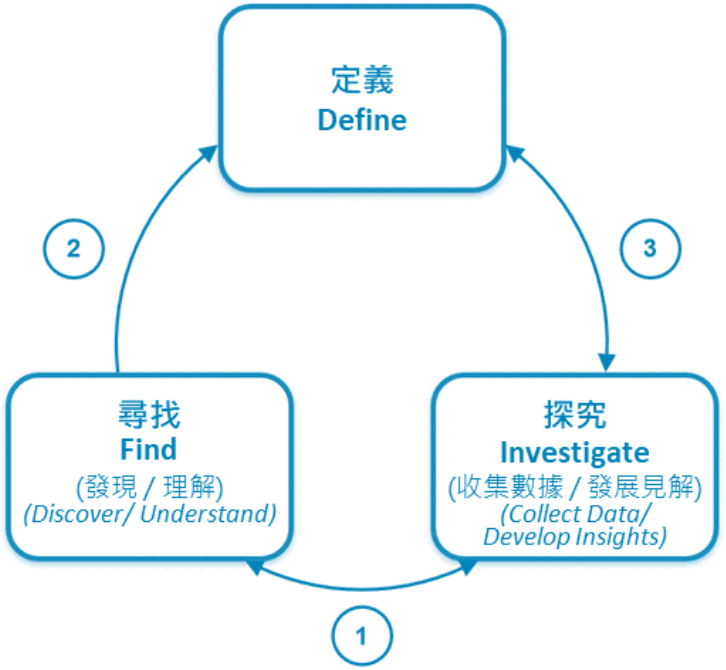
The ability to discover problems in the real life or raise questions in the digital world.

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

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

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 (App Inventor 編程環境)

第二步: CK (計算思維概念、實踐和態度)

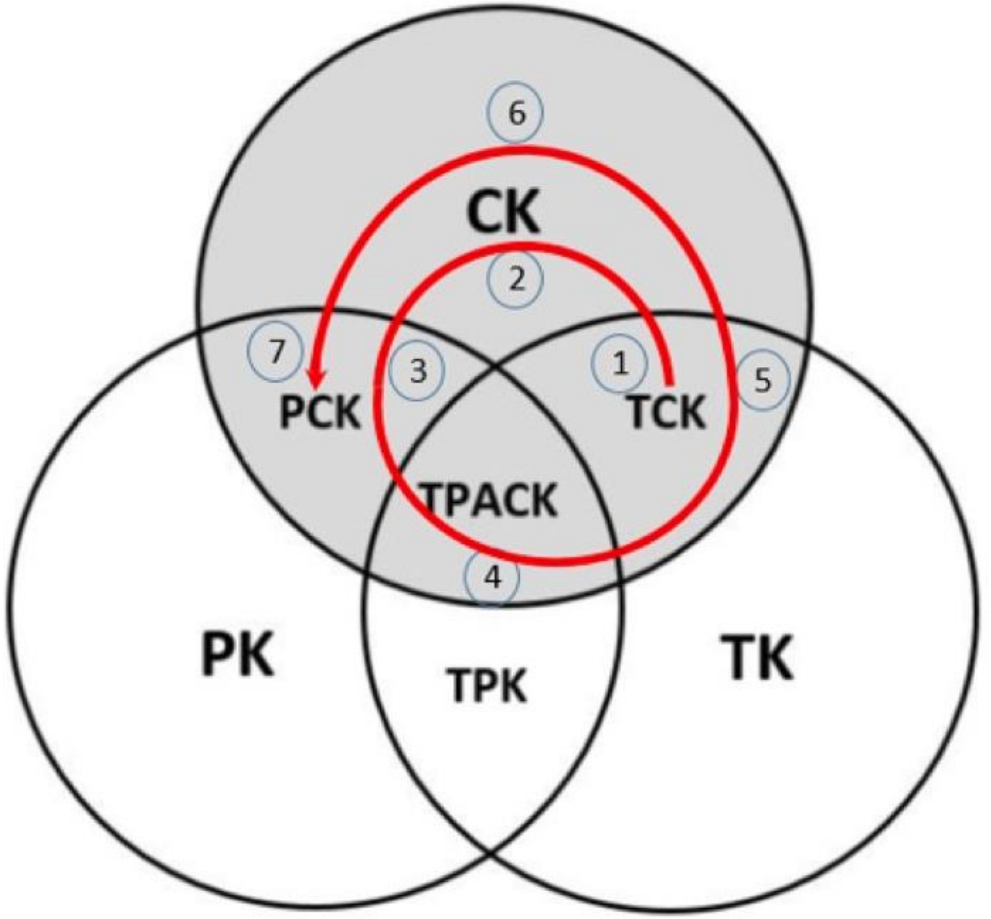
第三步: PCK (教學內容知識)

第四步: TPACK (科技教學學科知識)

第五步: TCK (數碼創意)

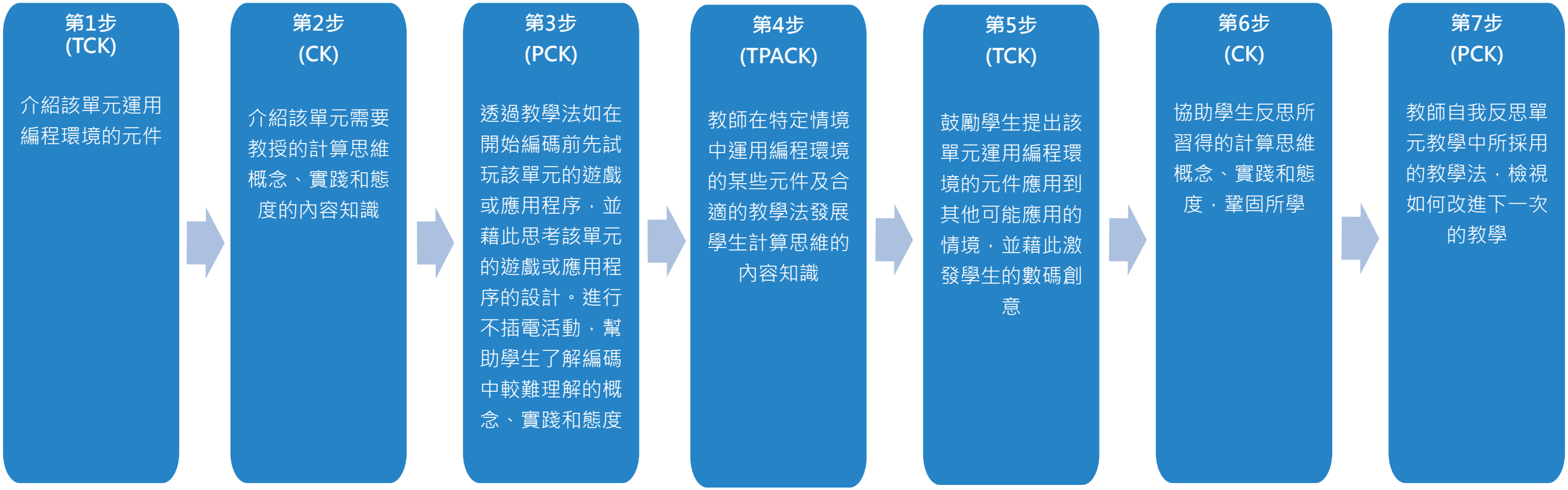
第六步: CK (總結計算思維概念、實踐和態度)

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



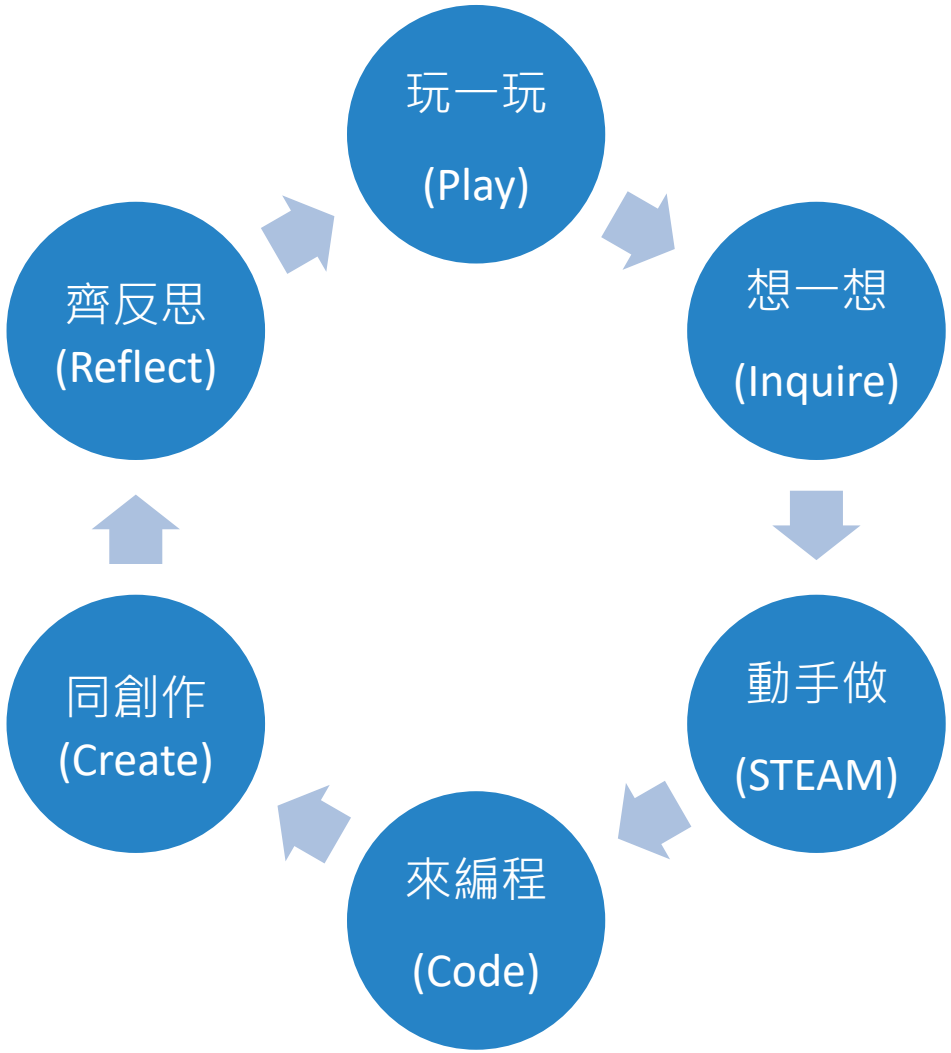
Teachers can flexibly arrange the steps to groom students' problem-solving skills and digital creativity.

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。

A six-step STEAM pedagogy

STEAM 教學法六步曲

STEAM 教學法六步曲



(Kong, 2024)

STEAM 教學法六步曲

第1步
玩一玩

供學生試玩單元的遊戲，讓他們初步理解系統的運作，為他們確立學習目標。

第2步
想一想

學生思考遊戲過程，拆解並探索系統的各個部分。

第3步
動手做

學生動手組裝不同部件來構建系統。通過解決問題，學生能培養因果推理及工程系統思維等解難能力。

第4步
來編程

學生進行編程，從而學習「序列」和「分支 / 選擇」等重要編程概念。

第5步
同創作

鼓勵學生提出該單元運用STEAM的概念應用到其他情境，並藉此激發學生的數碼創意。

第6步
齊反思

協助學生對同儕及自我進行反思，為對方的作品提供具建設性的回饋。

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： 數碼鋼琴 Digital Piano

單元示例：數碼鋼琴 — TPACK七步曲的運用

第1步 (TCK)

- 音樂播放器
- 按鈕
- 水平配置

第2步 (CK)

解決問題的過程：
- 問題定義
- 問題分析
- 算法設計
- 程序編寫

基本程序編寫結構：
- 事件
- 序列
- 命名

編程的概念與實踐：
- 設計、重用和混合程序/編碼
- 測試及除錯

態度
- 興趣
- 堅毅
- 創意
- 積極的價值觀

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序，並藉此思考該單元的遊戲或應用程序的設計。

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。

第5步 (TCK)

鼓勵學生提出該單元運用編程環境的元件應用內容以外的其他可能應用，並藉此激發學生的數碼創意。

例如：
- 改變鋼琴外型、或加入更多琴鍵？
- 音樂播放器還能播放什麼聲音？例如人聲、動物聲音、音樂或遊戲音效等。

第6步 (CK)

解決問題的過程：
- 問題定義
- 問題分析
- 算法設計
- 程序編寫

基本程序編寫結構：
- 事件
- 序列
- 命名

編程的概念與實踐：
- 設計、重用和混合程序/編碼
- 測試及除錯

態度
- 興趣
- 堅毅
- 創意
- 積極的價值觀

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。

- 利用範例（界面設計基本完成）來教學，能幫助學生掌握編程和運算思維嗎？
- 如沒有，學生有能力由零開始進行編程嗎？

(1) App Inventor 元件：音樂播放器

- 用於播放音效的多媒體元件，是一個**「不可見的元件」**。
- 在「畫面編排」及「程式設計」視窗中，可用「來源」屬性來指定音效來源，適合於播放較長的音效檔，如歌曲。

(1) App Inventor 元件：按鈕

- 使用者可透過點選、長按、放開按鈕等（事件）來觸發程序執行一些動作。
- 使用者也可以改變按鈕的外觀，如顏色、大小、形狀等。

(1) App Inventor 元件：水平配置

- 「水平配置」元件是用於「畫面編排」，可讓放進「水平配置」中的元件水平排列
- 如果希望元件上而下排列的話，則可用「垂直配置」元件。

(2) 學習目標

1. 加深對「畫面編排」內不同元件功能的理解；
2. 透過修改「按鈕」元件的屬性，改變其外觀以切合應用程式的需要；
3. 透過運用「音樂播放器」元件來製作一個能播放音效的程序；
4. 使用MIT App Inventor 的「程式設計」，編寫「數碼鋼琴」應用程式；
5. 運用計算思維實踐中的「設計、重用和混合程序 / 編碼」和「測試及除錯」；
6. 建立一個有趣的遊戲程式與朋友和家人分享，從而發展他們的計算思維能力和態度。

(2) 計算思維概念和實踐

主要學習元素	項目
算法	解決問題的過程： 問題定義、問題分析、算法設計、程序編寫 基本程序編寫結構： 事件、序列、命名 編程概念與實踐： 設計、重用和混合程序 / 編碼、測試及除錯

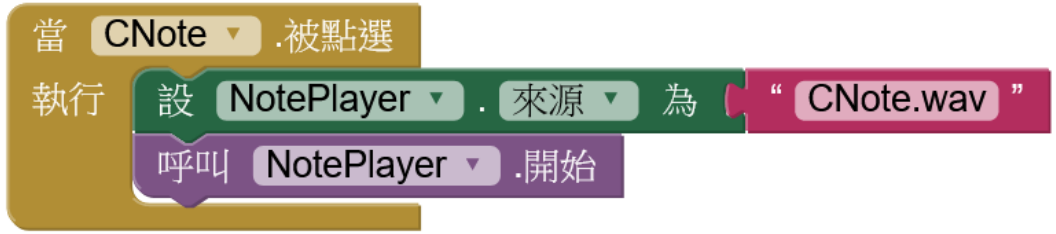
(2) 計算思維態度

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中，表現出堅毅及積極的態度；
3. 鼓勵學生展示創意，以改良他們的流動應用程式，或構思創建自己的專案。

(2) 計算思維概念和實踐

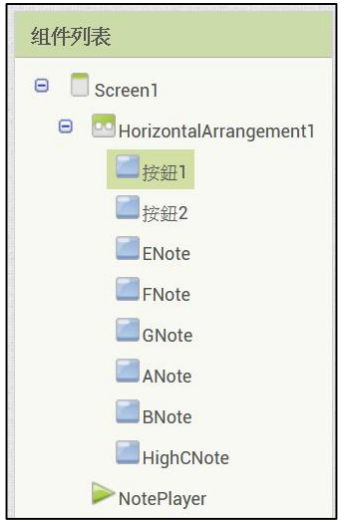
事件

序列



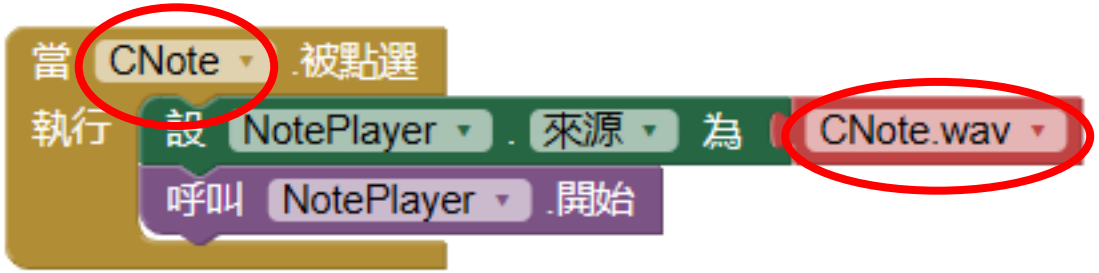
(2) 計算思維概念和實踐

命名

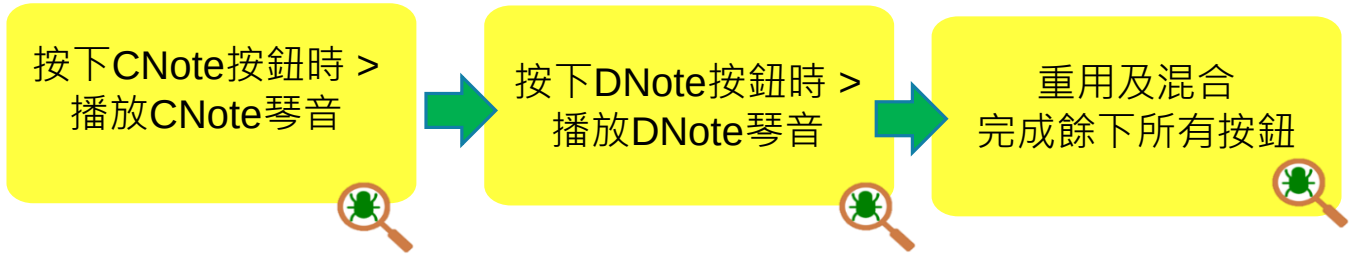


(2) 計算思維概念和實踐

重用及混合程序 / 編碼：

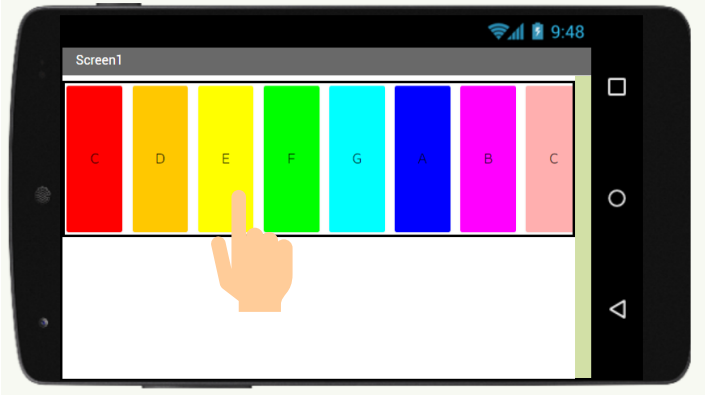


測試與除錯

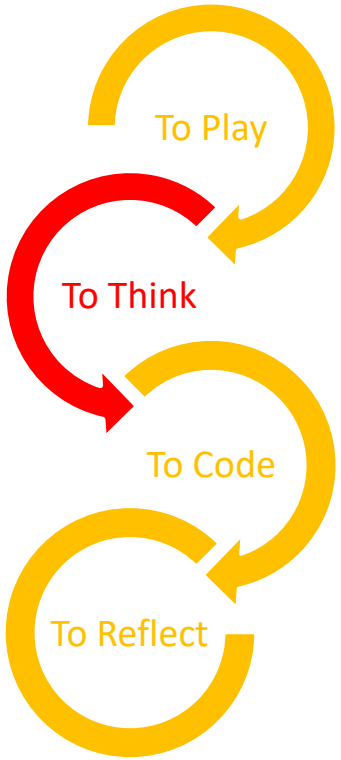


(3) 玩一玩 (To Play)

玩一玩 To Play:



(3) 想一想 (To Think)

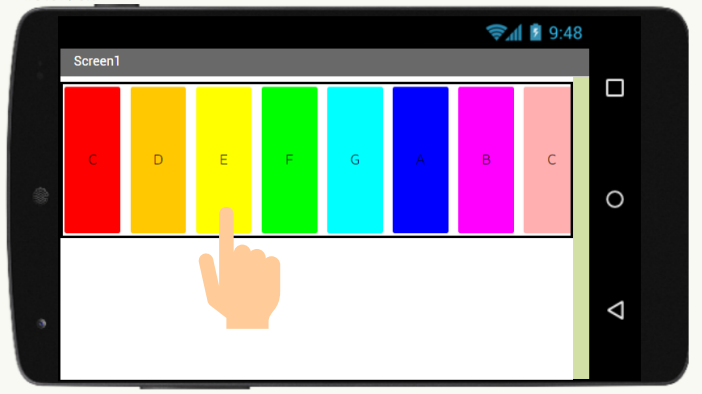


想一想 To Think:

- 當你試玩「數碼鋼琴」時，**遊戲呈現出甚麼**呢？
- 你認為這八個**琴鍵的元件**是甚麼？
- 哪個元件能**播放音效**？

(3) 想一想(To Think)

你認為這八個琴鍵的元件是甚麼？



Mentimeter

https://www.menti.com/alkr9rmpmq1c

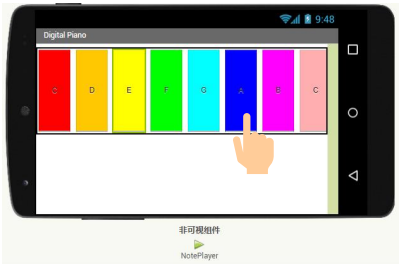
(3) 想一想 (To Think)

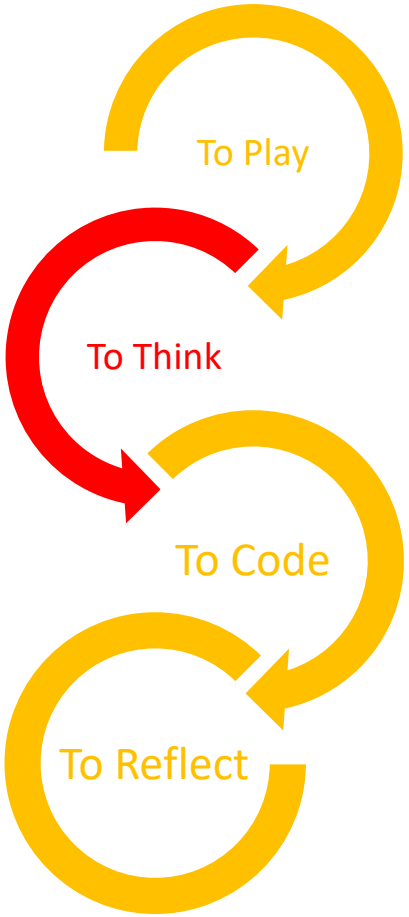
哪個元件能播放音效？

Mentimeter

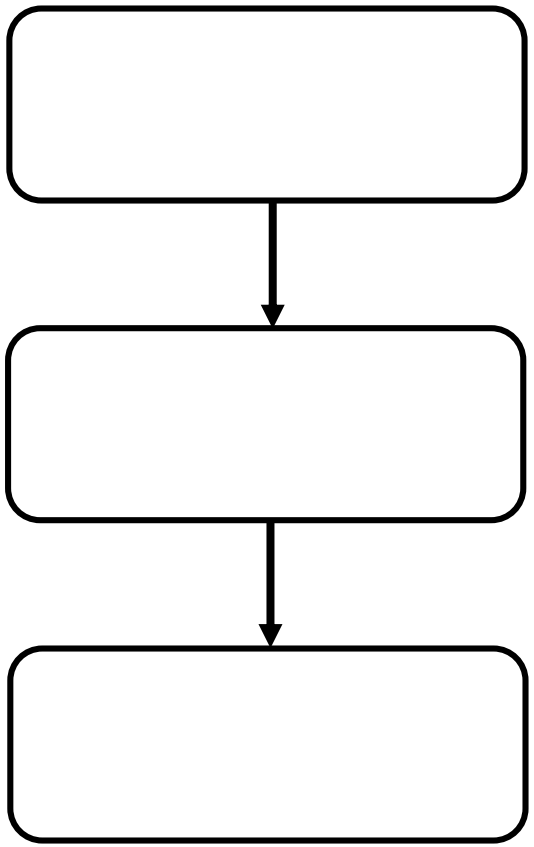
https://www.menti.com/alkr9rmpmq1c

(3) 想一想 (To Think)





A. 當使用者點擊琴鍵（按鈕）	B. 流動裝置播放相應琴音	C. 程序呼叫「音樂播放器」開始播放

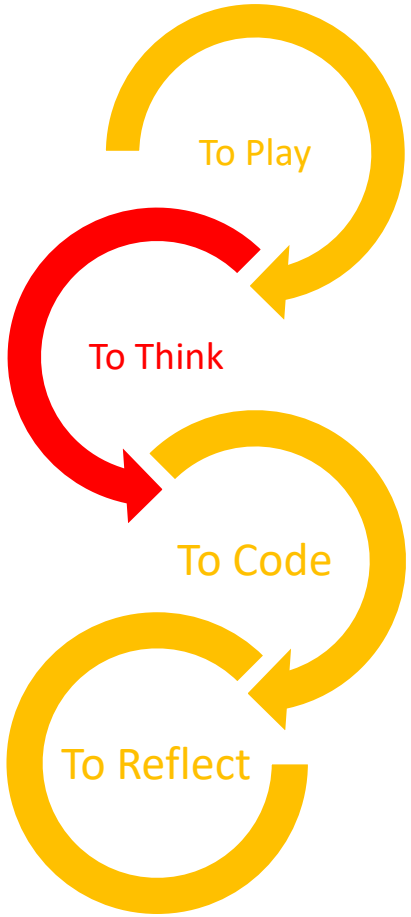


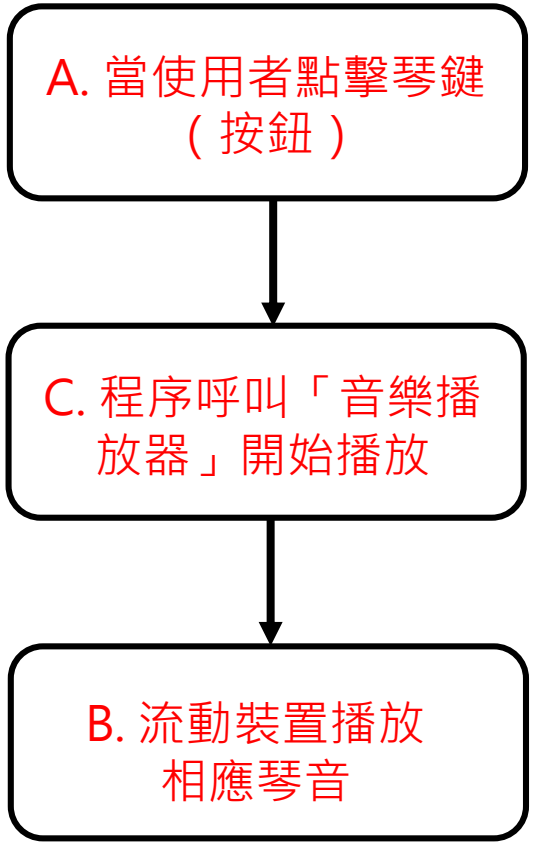
Mentimeter

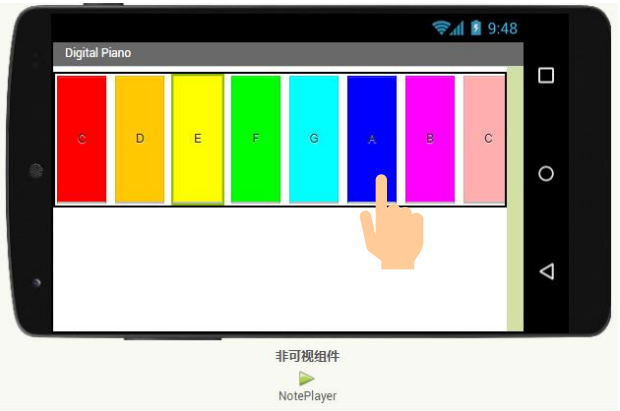
https://www.menti.com/alkr9rmpmq1c

(3) 想一想 (To Think)

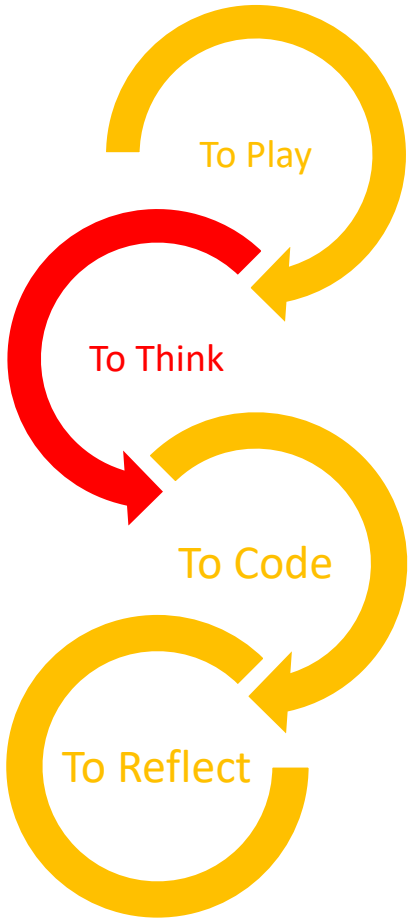
A. 當使用者點擊琴鍵（按鈕）	B. 流動裝置播放相應琴音	C. 程序呼叫「音樂播放器」開始播放





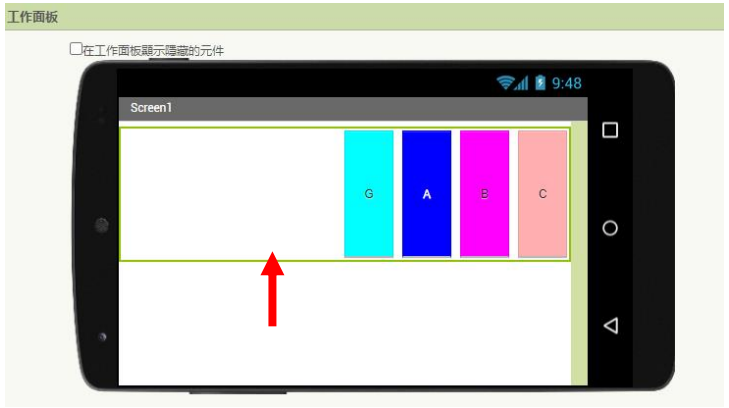


(3) 想一想 (To Think)：拆解問題

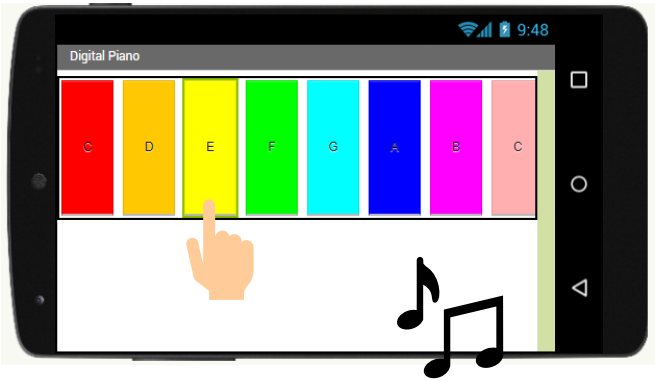


製作「數碼鋼琴」可以分為兩部分，逐一完成各任務就能製作完整的程式：

1. 以範本為基礎，於「畫面編排」介面加入「按鈕」及「音樂播放器」元件



2. 為所有「按鈕」編程：當按下不同琴鍵時，程式能播放相應琴音



(4) 科技教學學科知識 (TPACK)

1. 以範本為基礎，建立專案

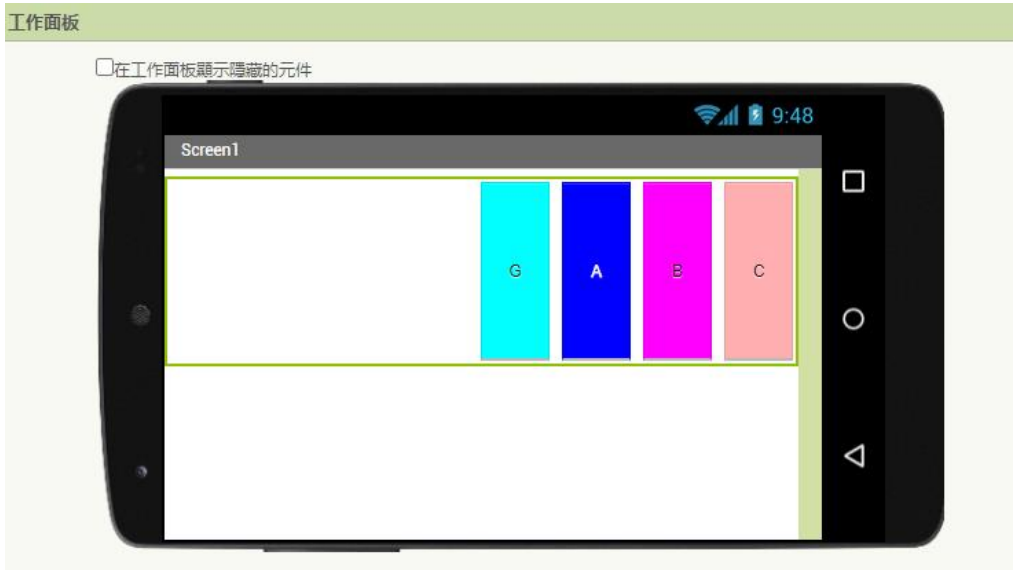
- 前往 MIT App Inventor 網站 https://ai2.appinventor.mit.edu 。
- 點擊連結下載「數碼鋼琴」範本專案到電腦。
  DigitalPiano_template.aia
- 點擊位於瀏覽器左上角的「專案」，然後選擇「匯入專案」。
- 選取範本專案「DigitalPiano_template」並打開，我們將以這範本為基礎，逐步完成屬於你的數碼鋼琴！

(4) 科技教學學科知識 (TPACK)

2. 從「畫面編排」認識程式的介面

- 打開範本專案後，你會看見部分鋼琴介面已完成！

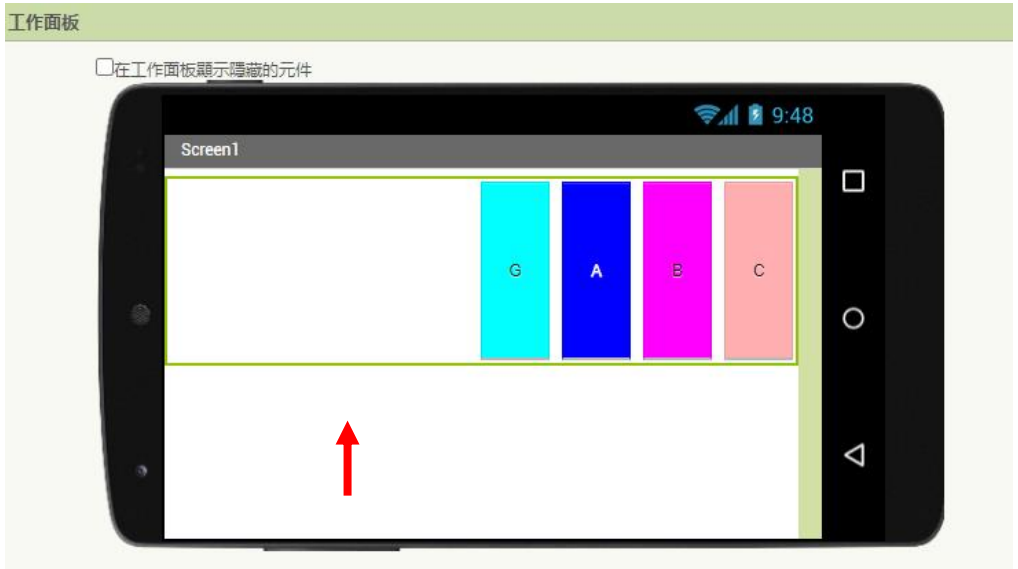
- 琴鍵已依照音符命名，例如 GNote、ANote 等。

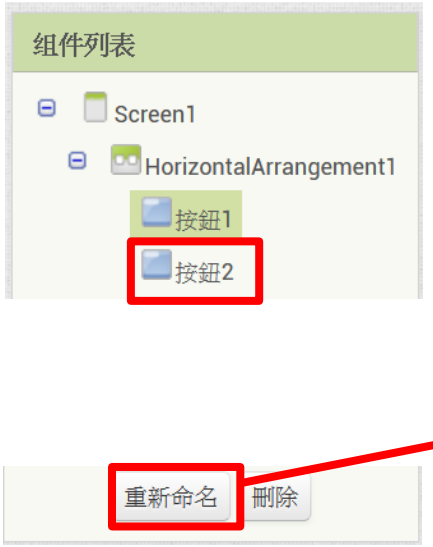


(4) 科技教學學科知識 (TPACK)

3. 加入新琴鍵（按鈕）

你有發現範本中的鋼琴像是缺少了部分琴鍵嗎？還記得如何加入「按鈕」嗎？





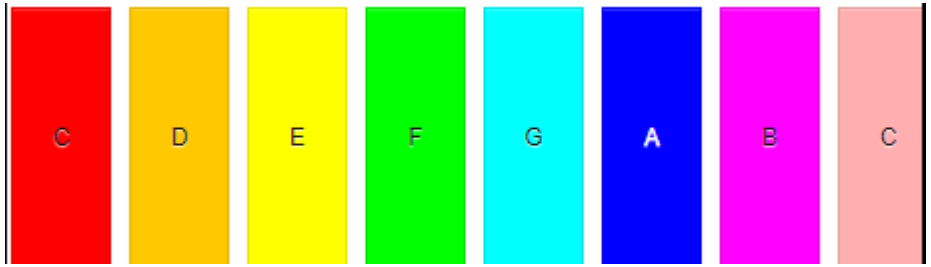
知識建立：命名

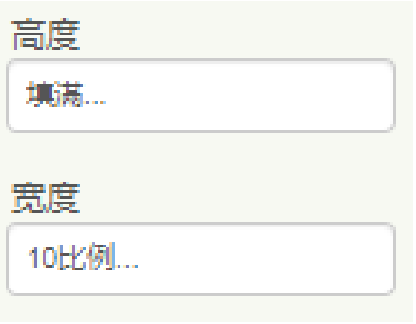
為元件命名，有助識別其功能及進行區別，例如識別哪個按鈕播放哪個音符的琴音。

(4) 科技教學學科知識 (TPACK)

修改新按鈕的元件內容（更改按鈕上顯示的文字、顏色、大小）

你有發現範本中的鋼琴像是缺少了部分琴鍵嗎？還記得如何加入「按鈕」嗎？





(4) 科技教學學科知識 (TPACK)

4. 加入音樂播放器 -「不可見的元件」

(4) 科技教學學科知識 (TPACK)

5. 為CNote按鈕（C琴鍵）編程

5. 為 CNote 按鈕（C 琴鍵）編程

還記得試玩「數碼鋼琴」時，當某音符的按鈕被點擊時，程序就會播放相應的琴音嗎？那我們現在需要為各個按鈕編程！

☐ 切換到「程式設計」編輯畫面，開始編程。

☐ 點選「積木」下的 CNote 按鈕，會顯示一系列指令方塊。

知識建立：事件

我們可運用不同的事件指令方塊來觸發 App Inventor 中的元件執行程序。
例如：當使用者按下 C 琴鍵（CNote）的按鈕元件時，就會播放 C 琴鍵相應的琴音。

☐ 要利用同一個「音樂播放器」來播放不同音符，**便要先設定它的音效來源**，令每按下一個琴鍵時，就播放相關的 wav 檔案。

☐ 拖放設 **NotePlayer.來源為 ANote.wav** 指令方塊。

☐ 把以上的指令方塊組合放在當 **CNote.被點選**指令方塊內。

☐ 點選 **ANote.wav** 並修改為 **CNote.wav**。

(4) 科技教學學科知識 (TPACK)

5. 為CNote按鈕（C琴鍵）編程

單元二
教學指引：附錄

☐ 要令程序能播放相應的琴音，我們需要運用 NotePlayer「音樂播放器」。

☐ 點擊 NotePlayer 元件，拖放 **呼叫 NotePlayer.開始** 指令方塊到 **當 CNote.被點選** 指令方塊之中。

知識建立：序列

這是編程的重要概念。程序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。

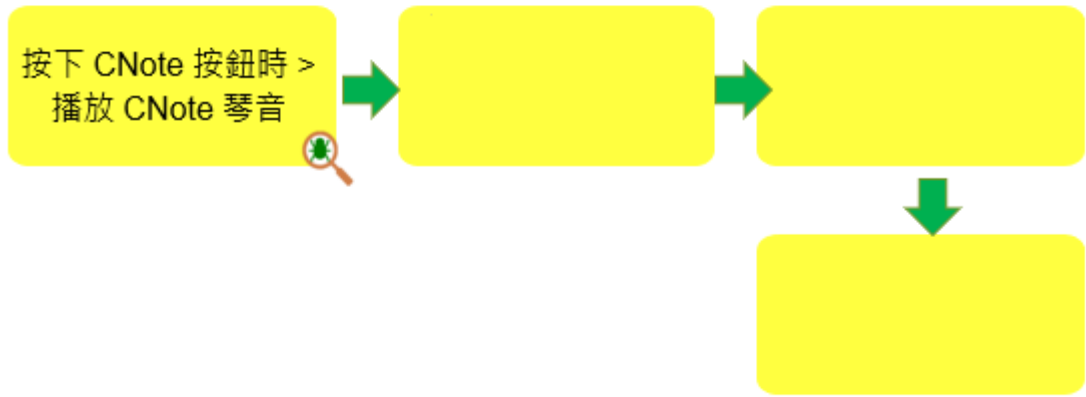
小提示

為琴鍵編程時，必須先加入音效「來源」，然後才呼叫「音樂播放器」開始播放，否則錯誤的次序會使程序無法正確執行，即程序無法播放聲音。

☐ 來測試你的程式吧！一邊修改程序，一邊作測試，找出不合理的地方。

測試及除錯

當按下 CNote 按鈕時，程序能否播放相應的 CNote 琴音？



(4) 科技教學學科知識 (TPACK)

6. 為DNote按鈕（D琴鍵）及餘下按鈕編程

單元二
教學指引：附錄

6. 為 DNote 按鈕（D 琴鍵）及餘下按鈕編程

☐ DNote 按鈕的程序就像 CNote 按鈕程序般運作，我們就複製及重用它的指令方塊吧！

☐ 以右鍵點擊 當 **CNote.被點選** 指令方塊，並在下拉式目錄中點選「複製程式方塊」。

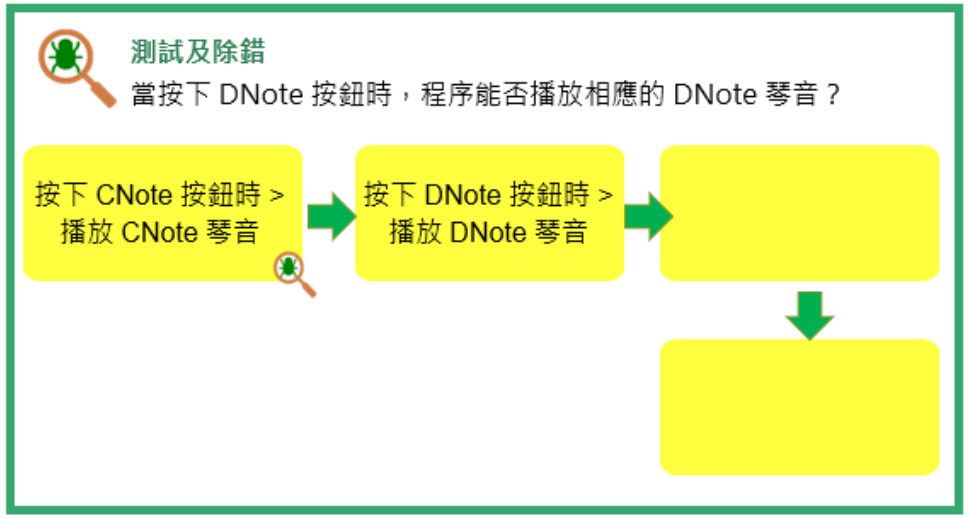
☐ 打開經複製而成的當 **CNote.被點選** 指令方塊，在下拉式目錄中把 CNote 改為 DNote。同時，修改 CNote.wav 為 DNote.wav。

小提示

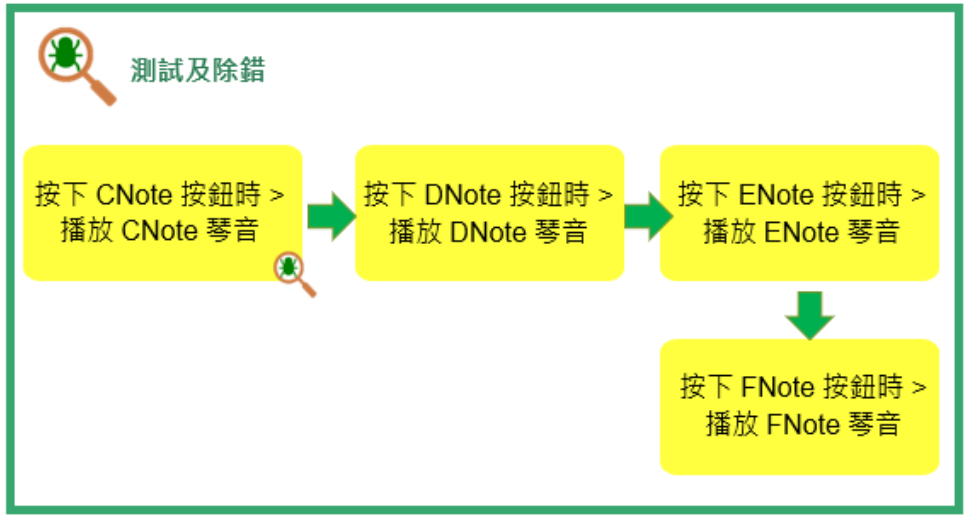
當複製一組指令方塊後，由於兩組指令方塊相同，左上角會出現紅色交叉提示，提醒你需要作出需改，否則程序無法執行。

謹記除了修改按鈕名稱，還要修改音效的來源！

☐ 來測試你的程式吧！一邊修改程序，一邊作測試，找出不合理的地方。



☐ 透過重用及混合以上指令方塊，完成餘下的琴鍵吧！

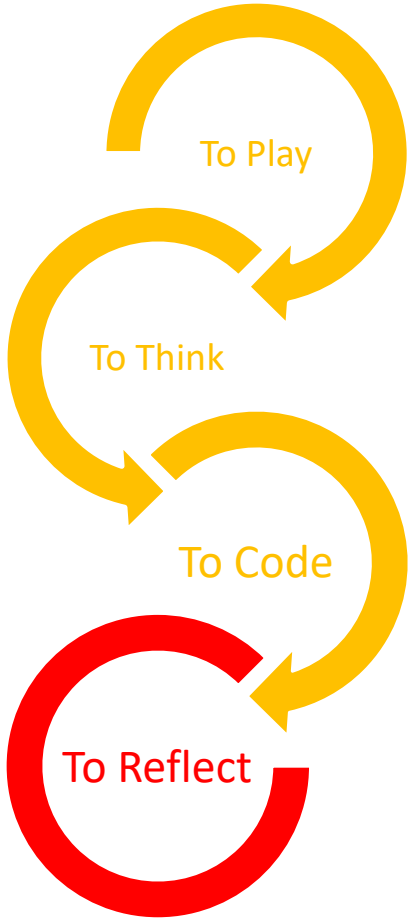


(4) 科技教學學科知識 (TPACK)

同創作：創建你的新專案

- 如果想令數碼鋼琴更有趣，會否**改變鋼琴的外型，或是加入更多琴鍵**？
- 試把你的概念畫出來吧！

(5) 科技內容知識反思，啓發數碼創意

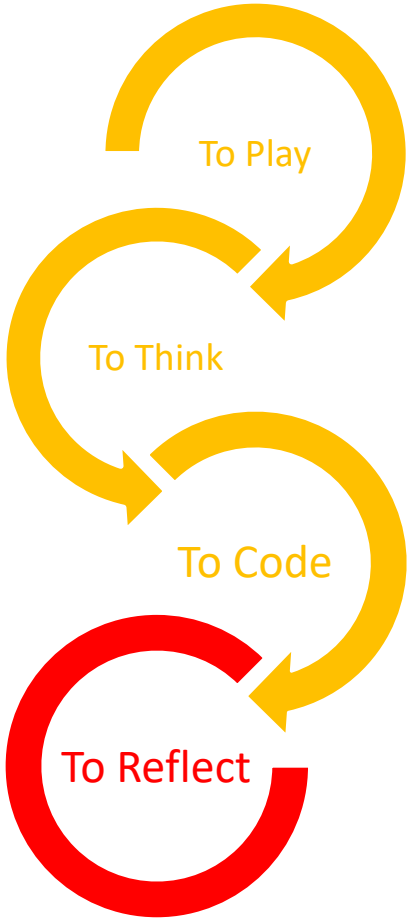


❖ 你能想到有應用「音樂播放器」至其他例子嗎？

Mentimeter

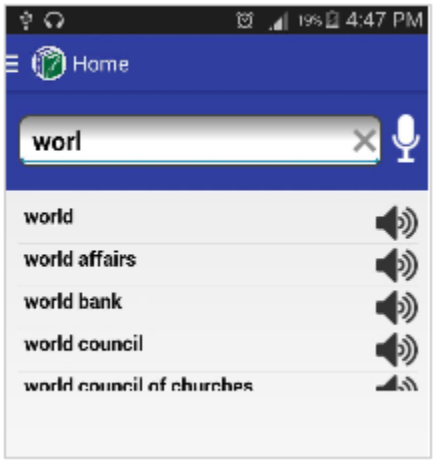
https://www.menti.com/alkr9rmpmq1c

(5) 科技內容知識反思，啓發數碼創意



❖ 你能想到有應用「音樂播放器」至其他例子嗎？

答：其他聲效，如人聲、動物聲音、音樂或遊戲音效等。



(6) 總結計算思維主要概念和實踐

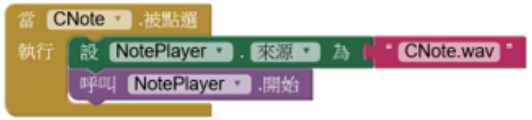
事件

我們運用事件指令方塊來觸發 App Inventor 中的元件執行程序。

例如：當 CNote 按鈕被點選時，就會觸發下一個程序的執行。

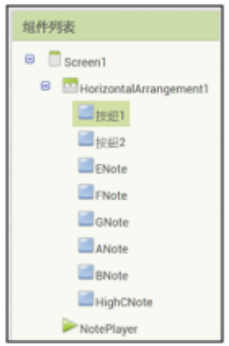
序列

這是編程的重要概念。程序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。例如，必須先加入音效「來源」，然後才呼叫音樂播放器，否則會無法播放聲音。



命名

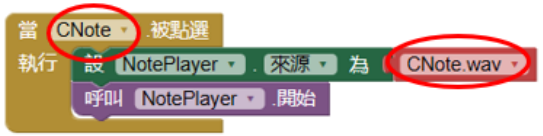
為元件命名有助識別其功能及進行區別。例如識別哪個按鈕播放哪個音符的琴音。



設計、重用和混合程序 / 編碼

運用重用和混合其他編碼對編程十分重要。

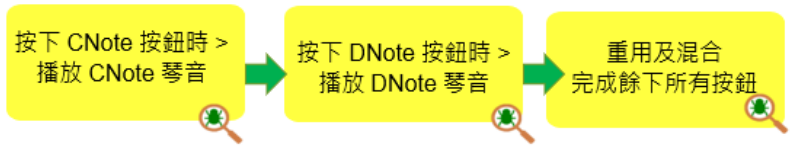
例如，我們可以重用和混合 CNote 按鈕事件的編碼，應用到其餘的按鈕。



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。

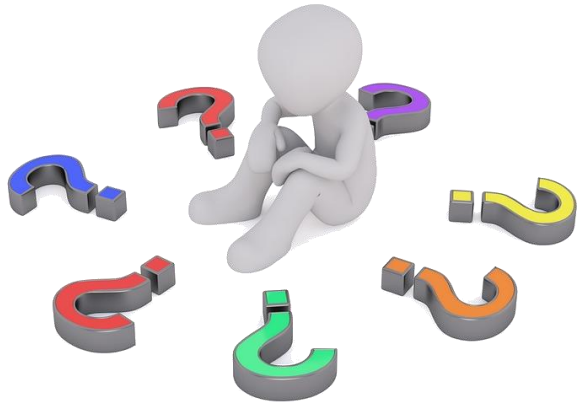
例如：測試按下 CNote 按鈕時，程序能否播放 CNote 琴音。測試電腦程序能否按原本的設計運作，然後找出錯誤並作修改。



(7) 教師自我反思教學法

1. 你認為利用範例（界面設計基本完成）來教學，能幫助學生掌握編程和運算思維嗎？為甚麼？

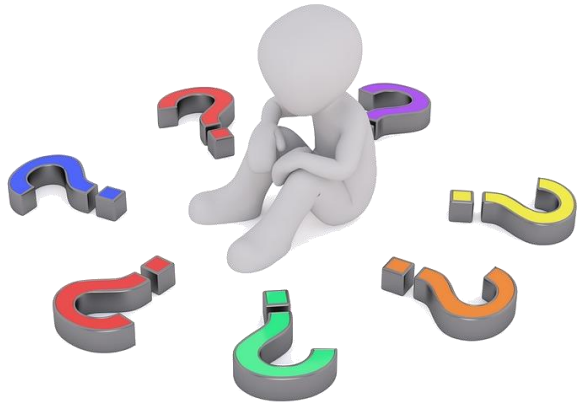
2. 這個範例沒有預載任何指令方塊，學生有能力由零開始進行編程嗎？為甚麼？

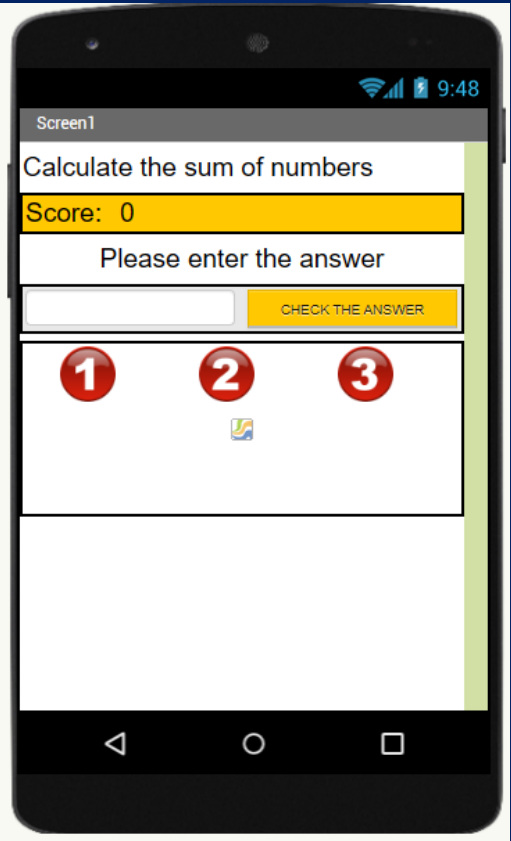


(7) 教師自我反思教學法

1. 你認為利用範例（界面設計基本完成）來教學，能幫助學生掌握編程和運算思維嗎？為甚麼？
   ☑ 集中精力編程，不必費神設計App的界面。
   ⊗ 缺乏機會讓學生自行創作並設計界面。

2. 這個範例沒有預載任何指令方塊，學生有能力由零開始進行編程嗎？為甚麼？
   ☑ 學生對編程已有一定的認識，只需簡單拖曳，先為一個琴鍵編程，再加以重用和整合，就能完成其餘的琴鍵。
   ⊗ 對於初學者來說，若範例內有多些指示會更好。





單元示例：加法遊戲 Addition Game

單元示例：加法遊戲 — TPACK七步曲的運用

第1步 (TCK)

- 畫布
- 圖像精靈
- 標籤

第2步 (CK)

解決問題的過程：

- 問題定義
- 問題分析
- 算法設計
- 程序編寫

基本程序編寫結構：

- 事件
- 序列
- 分支 / 選擇
- 命名
- 變量
- 程序

編程概念與實踐：

- 設計、重用和混合程序/編碼
- 測試及除錯

態度

- 興趣
- 堅毅
- 創意
- 積極的價值觀

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序，並藉此思考該單元的遊戲或應用程序的設計。
- 利用不插電活動，讓學生明白「變量」

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。

第5步 (TCK)

鼓勵學生提出該單元運用編程環境的元件應用內容以外的其他可能應用，並藉此激發學生的數碼創意。

例如：如何令數學題變成多樣化的加法遊戲？創造不同版本的玩法？

第6步 (CK)

解決問題的過程：

- 問題定義
- 問題分析
- 算法設計
- 程序編寫

基本程序編寫結構：

- 事件
- 序列
- 分支 / 選擇
- 命名
- 變量
- 程序

編程概念與實踐：

- 設計、重用和混合程序/編碼
- 測試及除錯

態度

- 興趣
- 堅毅
- 創意
- 積極的價值觀

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

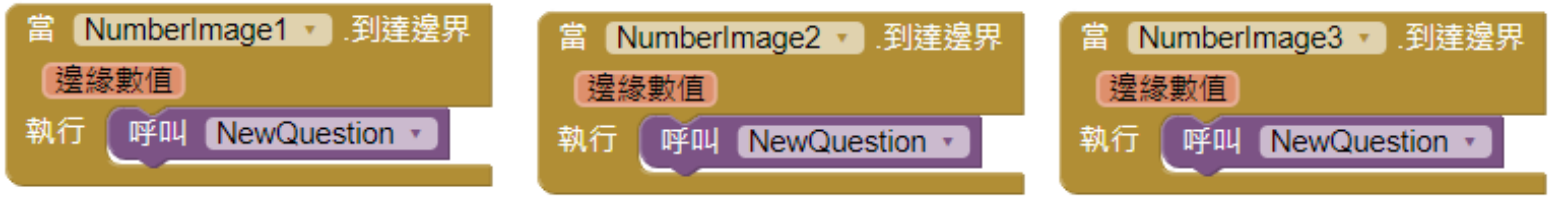
- 你會如何運用「加法遊戲」來提升同學創作其他學習程式的興趣?
- 你能否想到其他延伸的挑戰任務?
- 同學有否被啟發去運用及整合學科知識去製作屬於他們的程式?

(1) App Inventor 元件：畫布 (Canvas)

- 一個可觸控的平面長方形區域，可以在其上繪畫
- 讓球形精靈與圖像精靈在其中移動

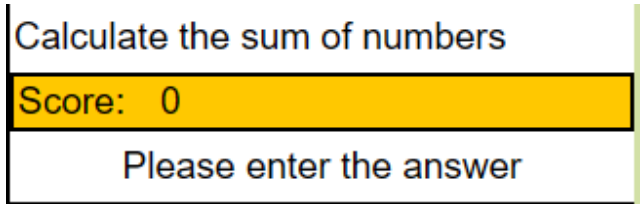
(1) App Inventor 元件：圖像精靈 (ImageSprite)

- 圖像精靈只能被放在畫布元件中
- 它可以回應觸碰與拖曳事件，與其他精靈(球形與其他圖像精靈)與畫布邊界產生互動
- 它的外觀可設定為某張圖片



(1) App Inventor 元件：標籤 (Label)

- 標籤可以顯示一段由文字屬性所指定的文字



(2) 學習目標

1. 加深對「畫面編排」內不同功能的理解；
2. 加深在手機編寫程序時控制屏幕上元件移動的理解；
3. 運用MIT App Inventor 的元件：如「文字輸入盒」來製作遊戲程式；
4. 展示對計算思維概念中的「程序」及「變量」的認識；
5. 提高對計算思維概念中的「事件」和「分支 / 選擇」的了解；
6. 建立一個與學科相關而有趣的遊戲程式與朋友和家人分享，從而有助提升他們對編程教育的學習成效。

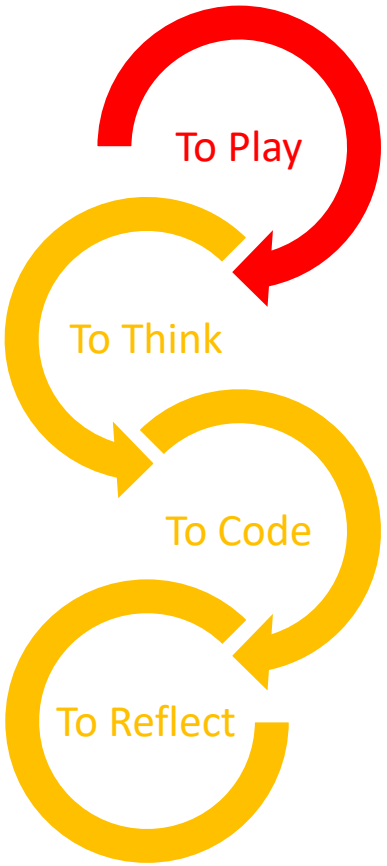
(2) 計算思維主要概念和實踐

主要學習元素	項目
抽象化	表達算法
算法	解決問題的過程： 問題定義、問題分析、算法設計、程序編寫 基本程序編寫結構： 事件，序列，分支 / 選擇，命名，變量，程序 編程概念與實踐： 設計、重用和混合程序/編碼、測試及除錯

(2) 計算思維態度

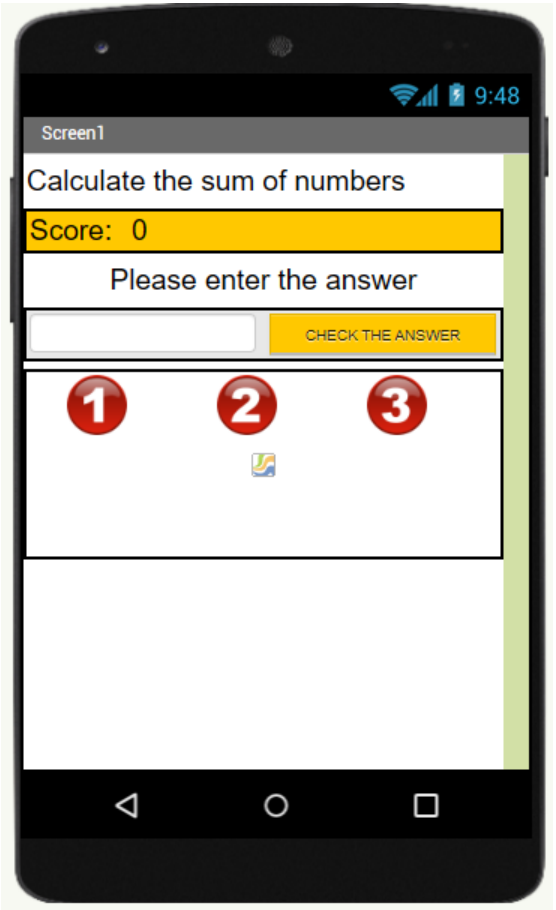
1. 培養學生對編程的興趣；
2. 鼓勵學生在測試及除錯的過程中，表現出堅毅及積極的態度；
3. 鼓勵學生展示創意及創造力，以構思、創建及改良自己的專案。

(3) 玩一玩 (To Play)

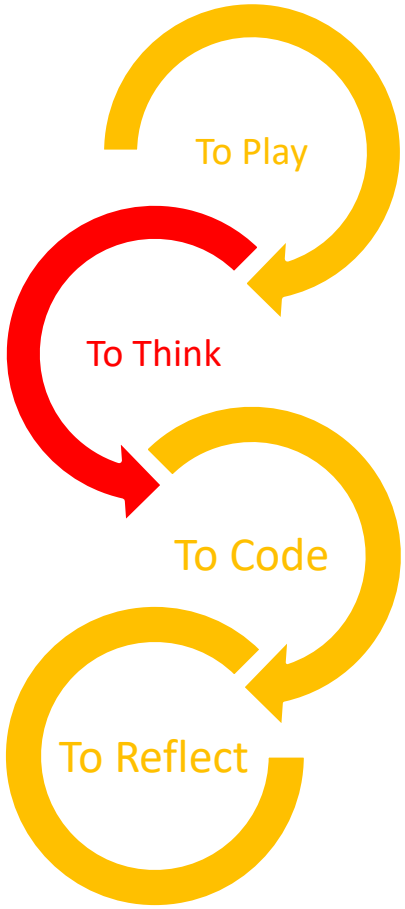


To Play:

- 試把程序中顯示的三個數字加起來。



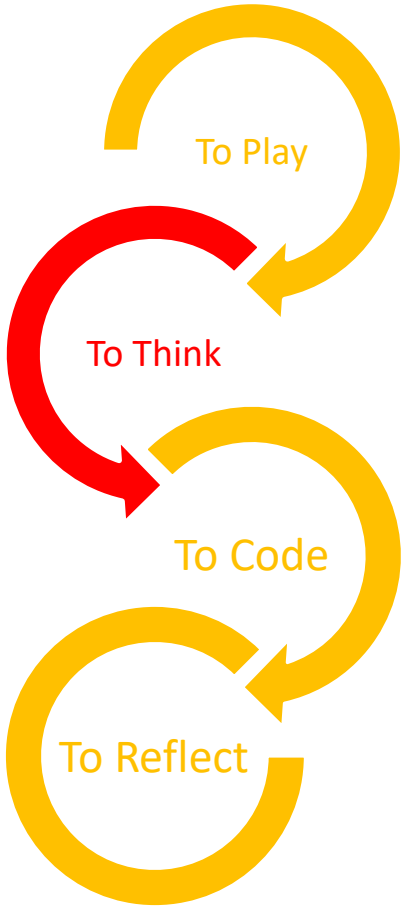
(3) 想一想 (To Think)



To Think:

- 遊戲開始時，畫面上有**多少個數字球**？它們**如何移動**？
- 當數字球**觸碰底部邊緣**時，發生甚麼事？
- 每次出現的三個數字會全部**相同數值**嗎？
- 把三個數字相加後：
  輸入**正確答案**，程式會發生甚麼事？
  輸入**錯誤答案**，又發生甚麼事？

(3) 想一想 (To Think)

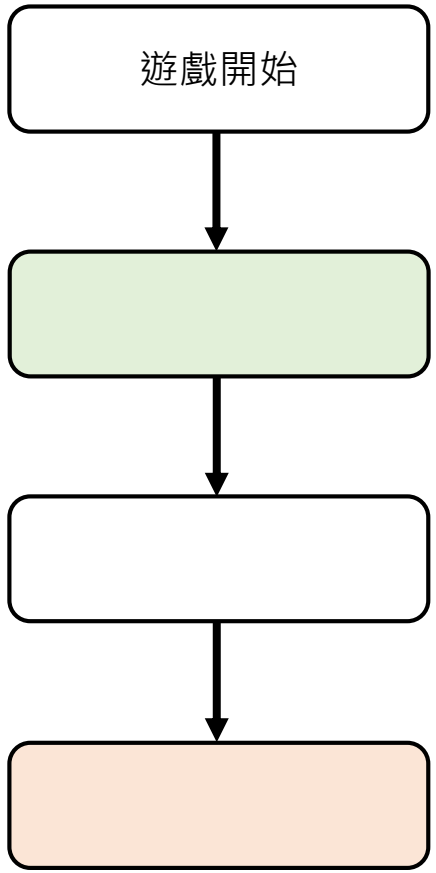


To Think:

- 遊戲開始時，畫面上有**多少個數字球**？它們**如何移動**？
  (共有三個數字球，由上而下移動。)
- 當數字球**觸碰底部邊緣**時，發生甚麼事？
  (它們會消失，然後新數值的數字球會再由上而下移動。)
- 每次出現的三個數字會全部**相同數值**嗎？(不相同。)
- 把三個數字相加後：
  輸入**正確答案**，程式會發生甚麼事？(分數會加一分，遊戲發出「yay」音效。)
  輸入**錯誤答案**，又發生甚麼事？(分數會減一分，遊戲發出「fail」音效。)

(3)拆解問題：流程圖

A. 使用者輸入答案	B. 生成一條新題目	C. 檢查答案

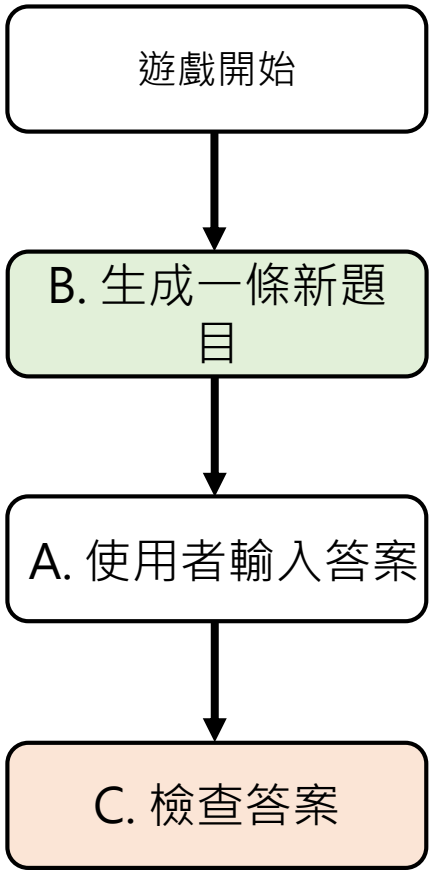


Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3)拆解問題：流程圖

A. 使用者輸入答案	B. 生成一條新題目	C. 檢查答案

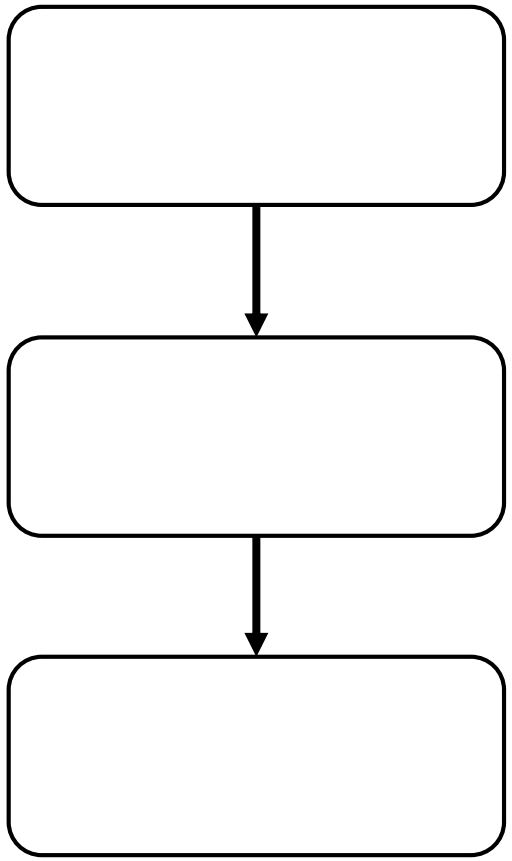


Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3)拆解問題：流程圖 - 生成一條新題目

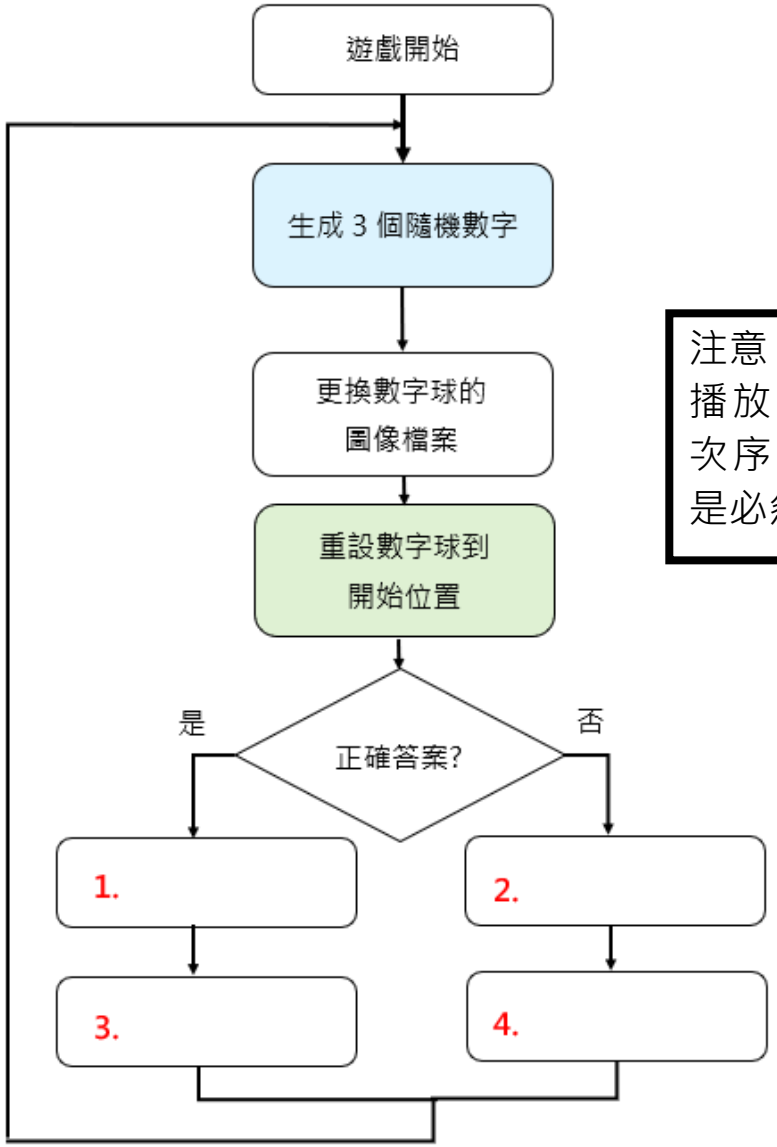
A. 生成 3 個隨機數字	B. 重設數字球到開始位置	C. 更換數字球的圖像檔案



Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3)拆解問題：流程圖



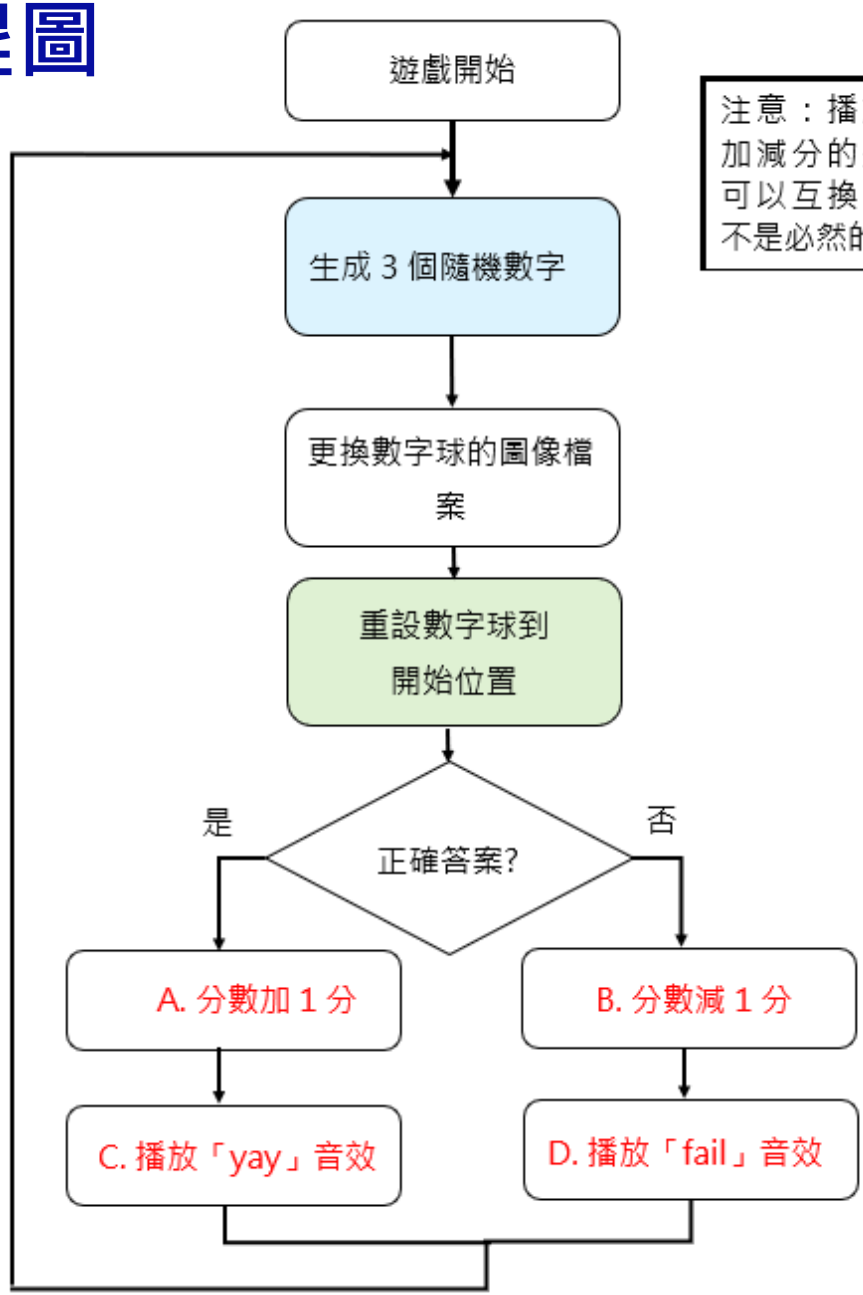
注意：
播放音效及加減分的先後次序可以互換，因為這不是必然的次序。

A. 分數加 1 分	B. 分數減 1 分
C. 播放「fail」音效	D. 播放「yay」音效

Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3)拆解問題：流程圖



注意：播放音效及加減分的先後次序可以互換，因為這不是必然的次序。

(4) 科技教學學科知識 (TPACK)

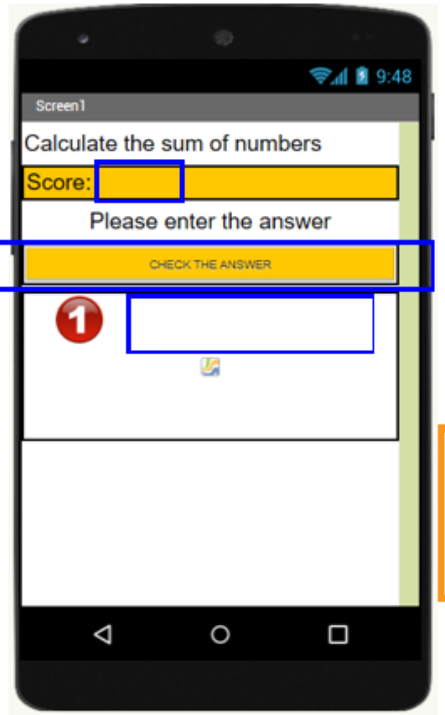
畫面編排 程式設計

1. 以範本為基礎，建立專案

☐ 前往 MIT App Inventor 並打開範例專案的 .aia 檔案。
Addition_Game_L1_template.aia

☐ 回想剛才試玩的「加法遊戲」，這個範本並未完成！以下藍色部份為缺少了的元件，它們分別是甚麼呢？請選出正確答案（答案可多於一項）。

☑ 標籤
☑ 圖像精靈
☐ 對話框
☑ 文字輸入盒



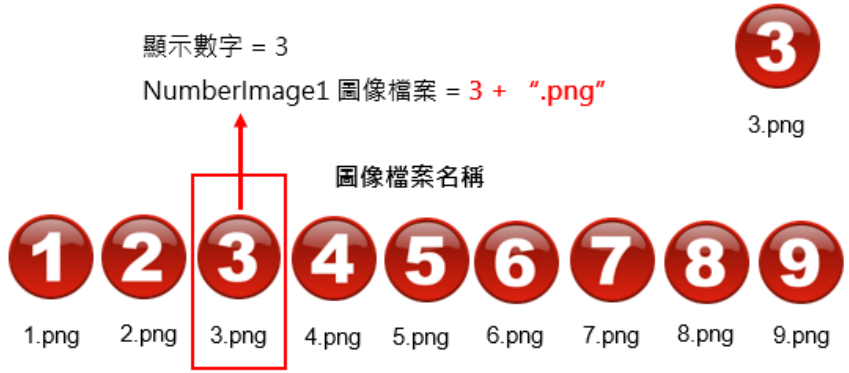
小提示
圖像精靈需要置於畫布上，否則會無法加入。

(4) 科技教學學科知識 (TPACK)

2. 從「畫面編排」加入及初始化兩個數字球

- 你有留意到畫面中數字球的元件是由圖像精靈(ImageSprite)配置嗎？如何能設置圖像精靈為數字球呢？每個數字球圖案都是由圖像檔案而成。它們可以在畫面編排或程序中改變為不同號碼的數字球。

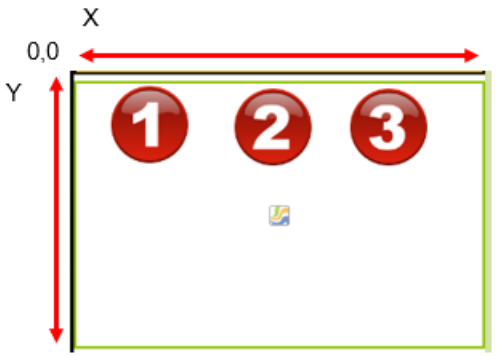
- 圖像檔案名稱是由檔案名稱及副檔案名稱(即圖片種類, 例如 .png)合併起來，程序就能按不同的數字而配對不同的圖像檔案！



- 留意數字球 1 的座標、「指向」方向和速度！試試修改為不同的數值，看看有甚麼分別？

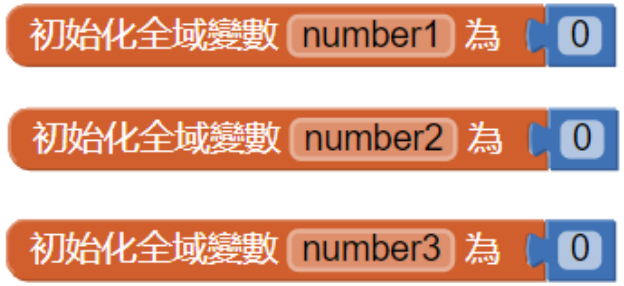
- 修改數字球的座標、「指向」方向和速度後，你能將遊戲的數字球的方向是從上而下嗎？我們可重溫單元四已學的知識內容，包括座標及指向的概念，在元件內容內修改相關數值。

NumberImage1 元件內容



(4) 科技教學學科知識 (TPACK)

3. 從「程式設計」建立數字球的變量



知識建立：變量

變量在編程用於儲存數值。每個變量都可設置名稱，而它每次只能儲存一個值，此值可以是數字或文字等，亦可在程序中變更新的數值。

(4) 科技教學學科知識 (TPACK)

4. 為生成一條新題目的程序編程

知識建立：程序 (procedure)

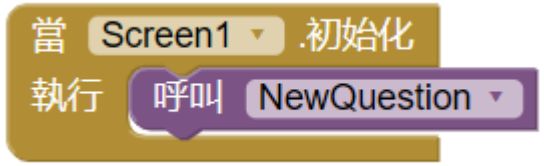
程序是把一連串編碼(指令方塊)組合起來，並加以命名。若要在程式的不同地方重複運用同一組指令方塊，可以呼叫程序的名稱，取代多次重覆的指令。在這加法遊戲中，我們運用了名為「NewQuestion」的程序。

重用及混合numberImage1的指令方塊，完成numberImage2及numberImage3部份

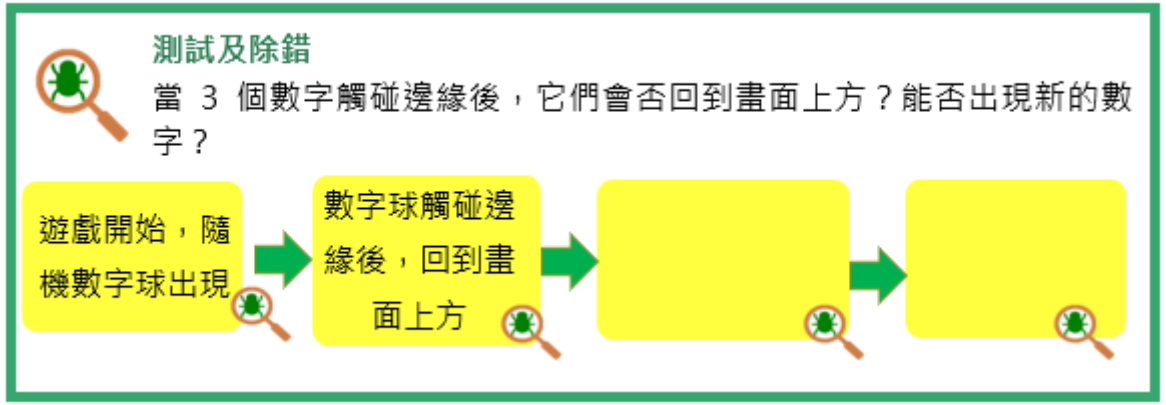
(4) 科技教學學科知識 (TPACK)

5.為觸發產生一條新題目的程序編程

i. 遊戲開始時



ii. 當數字球觸碰底部邊緣後

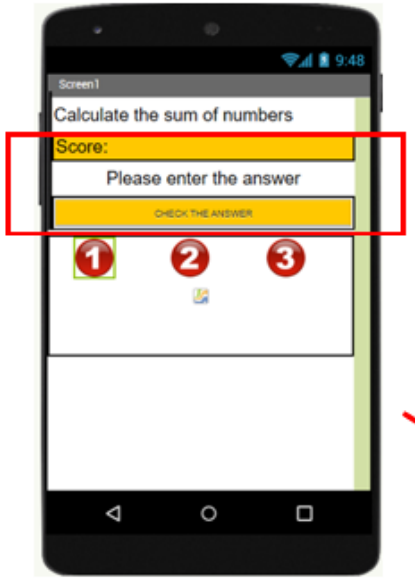


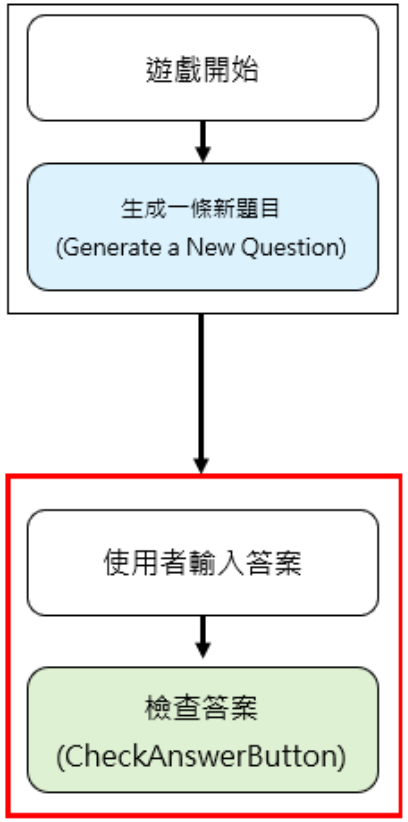
(4) 科技教學學科知識 (TPACK)

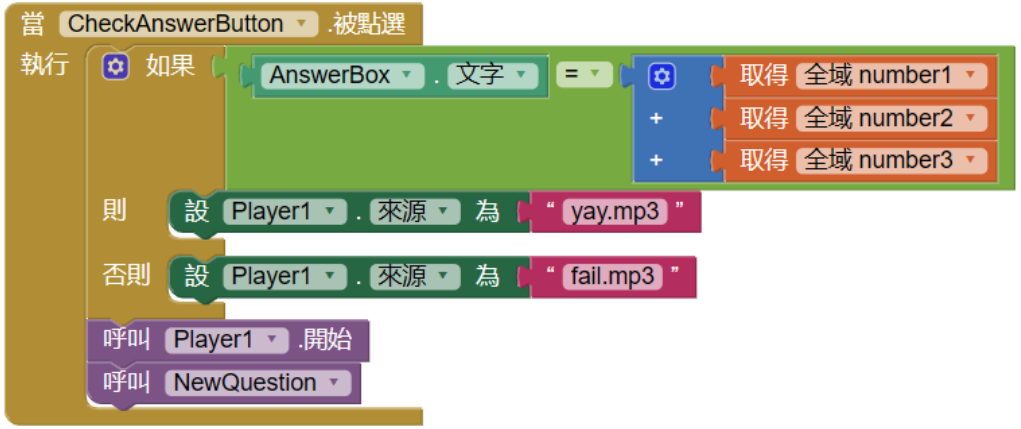
6. 為檢查答案「CheckAnswerButton」按鈕編程

a. 從「畫面編排」加入「AnswerBox」儲存答案

b. 從「程式設計」檢查答案







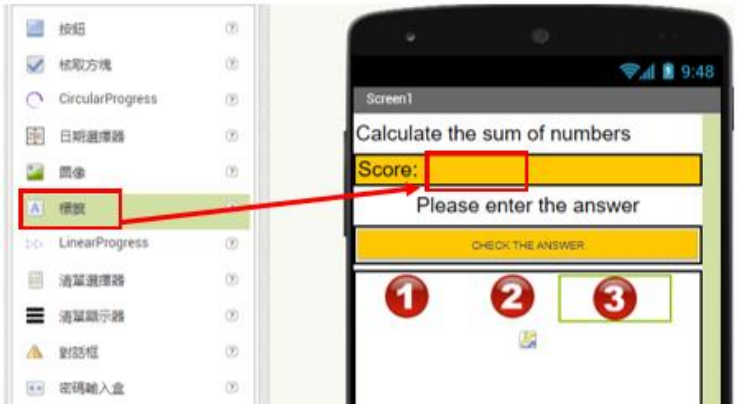
測試及除錯

如果輸入的答案正確，能播放 yay.mp3 音效檔嗎？如果答案錯誤，則播放 fail.mp3 音效檔。

(4) 科技教學學科知識 (TPACK)

7. 為得分編程

a. 從「畫面編排」加入「分數」的標籤



b. 從「程式設計」加減分數

測試及除錯

輸入答案後，程序會因應答案正確與否，而加分或減分嗎？

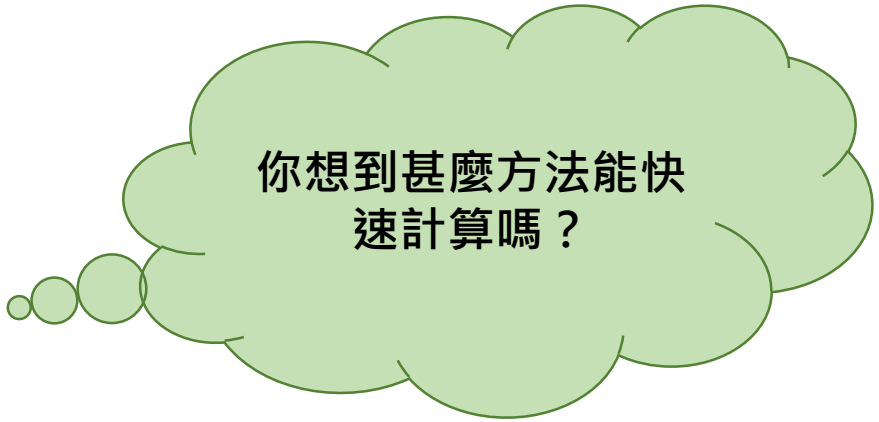
遊戲開始，隨機數字球出現 → 數字球觸碰邊緣後，回到畫面上方 → 輸入答案後，程序能核對答案並發出相應音效 → 程序能加分及減分

(4) 科技教學學科知識 (TPACK)

加法遊戲 — 挑戰任務

於本挑戰任務，你會再改進遊戲，令這「加法遊戲」能讓低年級同學練習「合十」的算法。

2 + 5 + 8 = ?
4 + 7 + 6 = ?
1 + 3 + 9 = ?



(4) 科技教學學科知識 (TPACK)

你想到甚麼方法
能快速計算嗎？

加法遊戲 — 挑戰任務

於本挑戰任務，你會再改進遊戲，令這「加法遊戲」能讓低年級同學練習「合十」的算法。

每題數其中兩個加起來會等於１０，如「2和8」、「4和6」、「1和9」都是「10的組合」，我們先加可組成10的數字，便可以快速計算到各題的答案。

2 + 5 + 8 = ?
4 + 7 + 6 = ?
1 + 3 + 9 = ?

(4) 科技教學學科知識 (TPACK)

完成「合十」的組合

小提示

把變數 number3修改成 **10 減去變數 number1**，就可以令它們「合十」。

(5) 科技內容知識反思，啓發數碼創意

提問學生想運用「加法遊戲」學到的來創造什麼？

✓ 例如：你想創造什麼新遊戲？

如何令數學題變成多樣化的加法遊戲？

創造不同版本的玩法？

Mentimeter

https://www.menti.com/alkr9rmpmq1c

(6) 總結計算思維主要概念和實踐

事件

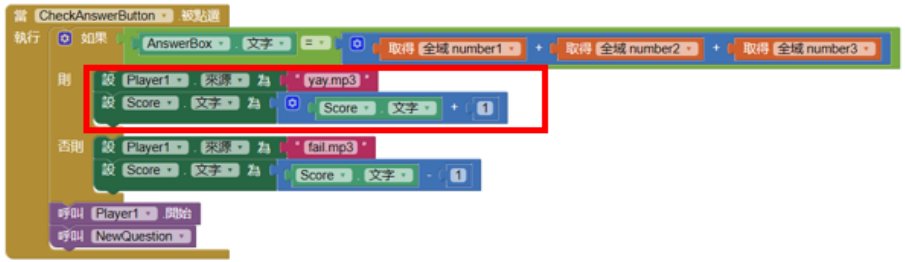
我們運用事件指令方塊來觸發 App Inventor 程序的執行。

例如：當 Screen1 被初始化，然後它會觸發程序產生新題目。

序列

這是編程的重要概念。程序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。

例如，如調轉播放音效跟更改分數的次序，會影響程序執行這兩行動的次序。



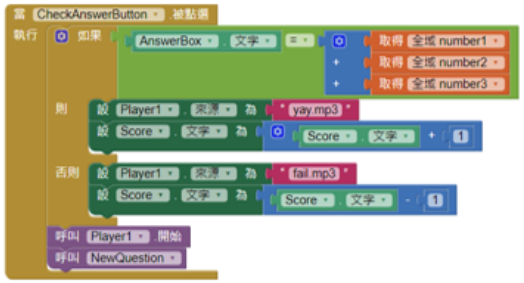
分支 / 選擇

我們在編程運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，它告訴程序當條件為真成立時，「那麼」應該做甚麼。

例如：[條件句]「如果」輸入的答案正確

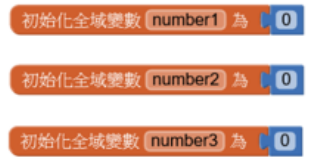
[結果] 「則」播放 yay 音效，分數加一分

「否則」播放 fail 音效，分數減一分



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。



(6) 總結計算思維主要概念和實踐

程序

程序是把一連串編碼組合起來並命名的指令方塊。若要在程式的不同地方重複運用同一組指令方塊，程序便十分實用。在這加法遊戲中，我們運用了名為「NewQuestion」的程序。

設計、重用和混合程序 / 編碼

我們在編程社群中，運用重用和混合其他編碼十分重要。

例如，我們可以重用和混合 number1 的編碼，應用到 number2 及 number3 的程序。

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。

例如：數字球會因應不同情況回到畫面上方。然後程序於玩家輸入答案後，能核對答案並發出相應音效、加分及減分。測試電腦程序能否按原本的設計運作，然後找出錯誤並作修改。

抽象化

抽象化是一個過程，將複雜的問題的描述簡化，方法是隱藏問題中詳細的描述，而保留重要的相關資訊。例如，我們想將一般的單位數的加法遊戲，變成一個具「合十」組合的加法遊戲，就是一個抽象的描述。

(6) 總結計算思維主要概念和實踐

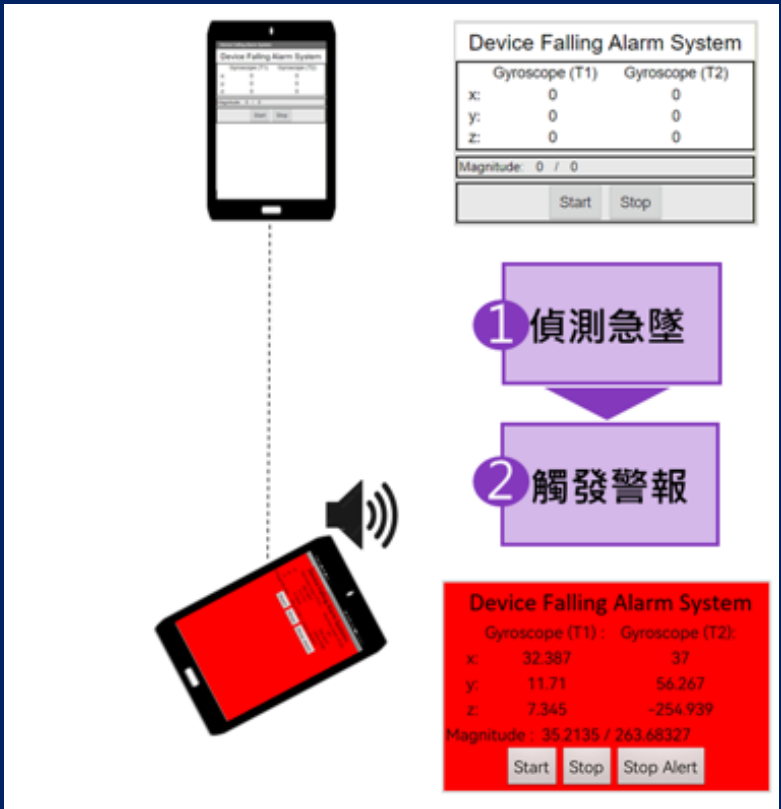
算法

算法是一組指示電腦來執行的指令，電腦藉著執行此特定的指令來完成特定的任務。例如，我們的特定任務是將普通的單位數加法遊戲，變成一個具「合十」組合的加法遊戲。遊戲原本的設計是有三個隨機單位數的數字球，我們加入新的指令，把第 3 個數字球設置為 10 減第 1 個數字球的值。

編碼例子：我們根據上述的算法設計來完成程序編寫，以下是具「合十」組合的加法遊戲編碼。

(7) 教師自我反思教學法

- 你會如何運用「加法遊戲」來提升同學創作其他學習程式的興趣?
- 你能否想到其他延伸的挑戰任務?
- 同學有否被啟發去運用及整合學科知識去製作屬於他們的程式?



單元示例：裝置急墜警報系統
Device Falling Alarm System

單元示例：裝置急墜警報系統 — TPACK七步曲的運用

第1步 (TCK)

- 計時器
- 陀螺儀感測器

第2步 (CK)

解決問題的過程：

- 問題定義
- 問題分析
- 算法設計
- 程序編寫

基本程序編寫結構：

- 事件
- 序列
- 分支 / 選擇
- 循環
- 命名
- 變量

編程概念與實踐：

- 設計、重用和混合程序/編碼
- 測試及除錯

態度

- 興趣
- 堅毅
- 創意
- 積極的價值觀

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序，並藉此思考該單元的遊戲或應用程序的設計。
- 運用「裝置急墜警報系統」記錄實驗數據：測試並記錄急墜時陀螺儀 x-軸、y-軸、z-軸的角速度和量值的數值。

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。

第5步 (TCK)

鼓勵學生提出該單元運用編程環境的元件應用內容以外的其他可能應用，並藉此激發學生的數碼創意。

例如：有什麼方法可以幫助我們自動偵測長者是否跌倒了？當長者跌倒時，我們希望系統能夠做些什麼？你能想像出一個方法，使系統能夠自動發出求救訊號嗎？

第6步 (CK)

解決問題的過程：

- 問題定義
- 問題分析
- 算法設計
- 程序編寫

基本程序編寫結構：

- 事件
- 序列
- 分支 / 選擇
- 循環
- 命名
- 變量

編程概念與實踐：

- 設計、重用和混合程序/編碼
- 測試及除錯

態度

- 興趣
- 堅毅
- 創意
- 積極的價值觀

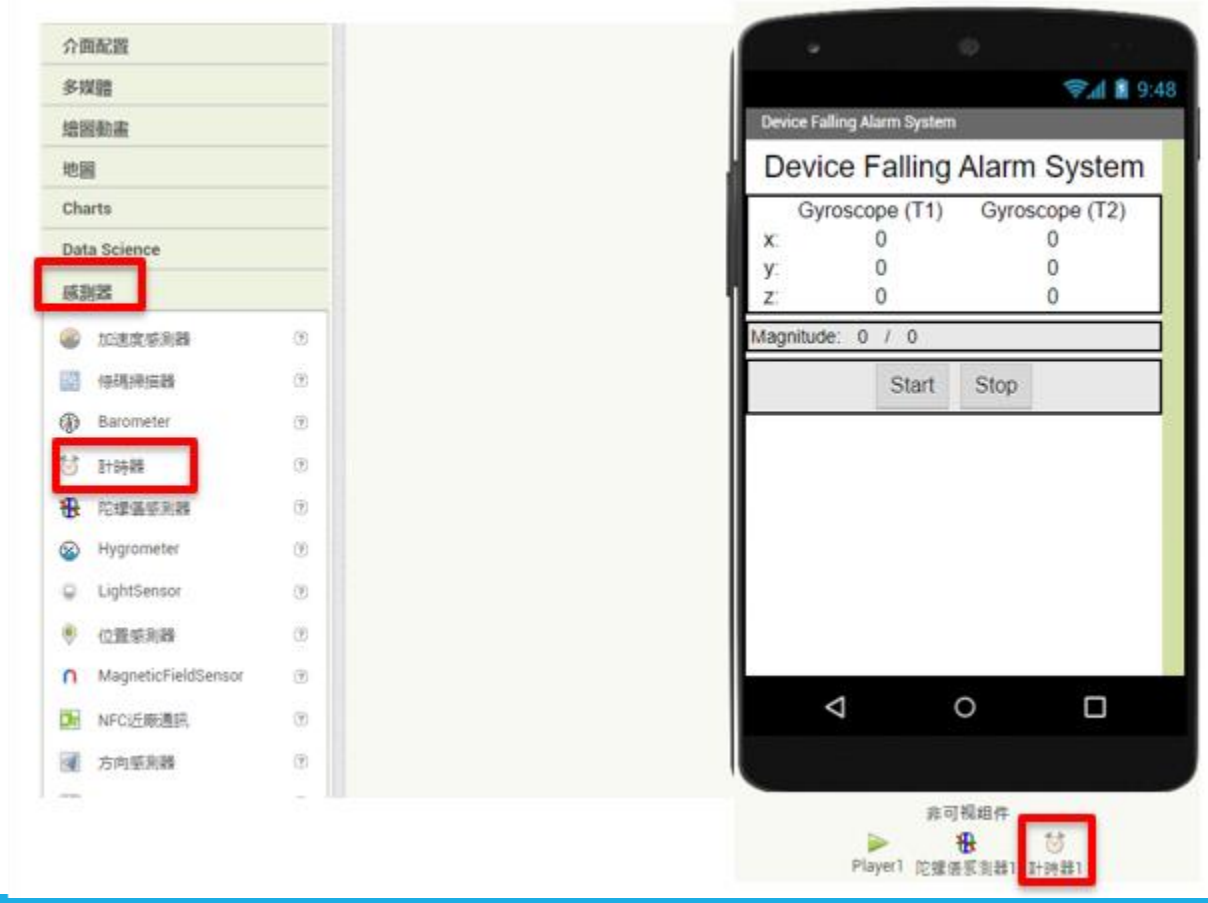
第7步 (PCK)

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

- 假設每秒量值的數值差超過120，則可能判定裝置正在急墜，並觸發警報。您覺得這樣的設定合適嗎？
- 你能否想到其他延伸的挑戰任務?
- 同學有否被啟發去運用及整合學科知識去製作屬於他們的程式?

(1) App Inventor 元件：計時器 (Clock)

- 運用裝置內部時鐘提供即時時間。
- 它可以定期設定的時間間隔觸發計時器並執行時間計算、操作和轉換。



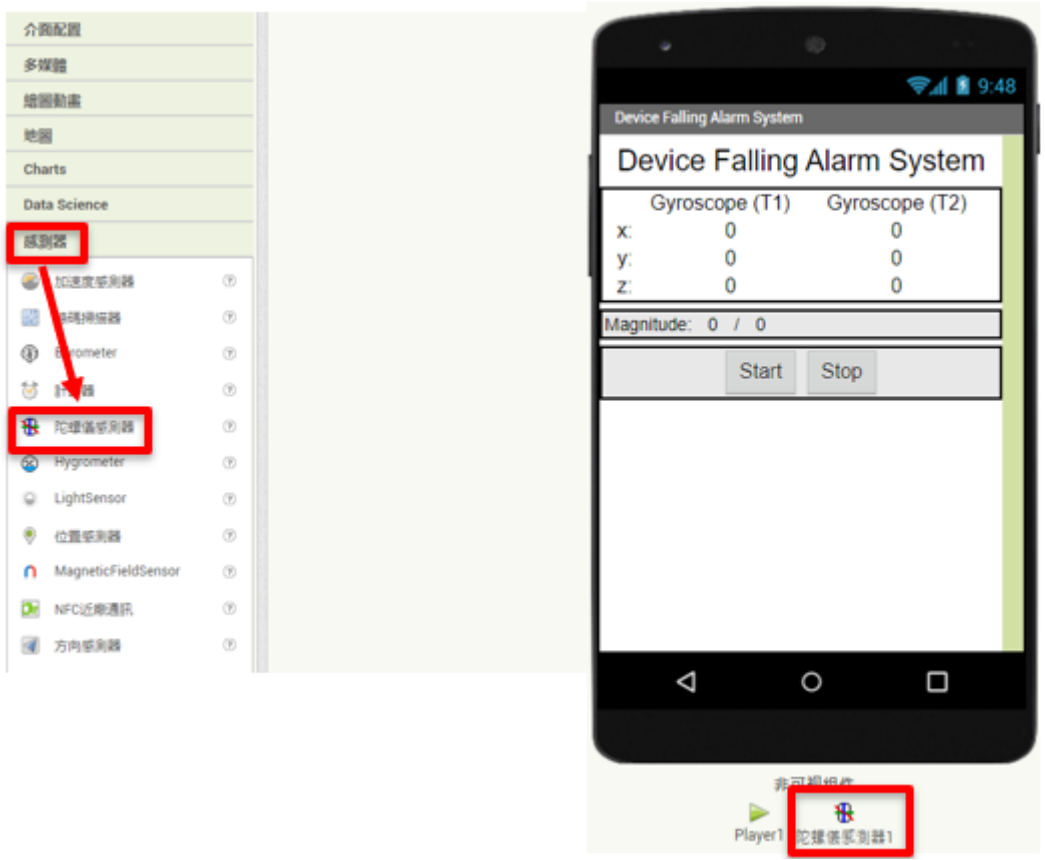
持續計時
即使 App Inventor 程式沒在畫面前端，計時器還是會一直觸發並運行程序。

啟用計時
當計時器已啟用，才可觸發計時器並運行程序。

計時間隔
單位是**毫秒**，有別我們常用的秒或分鐘，毫秒是比秒更小的單位。例如，等待 1000 毫秒 = 1 秒之後，等待 5,000 毫秒 = 5 秒之後。

(1) App Inventor 元件：陀螺儀感測器 (GyroscopeSensor)

- 是測量傾斜運動的工具，用於測量和監測物體的旋轉或傾斜。
- 它通常由一個旋轉的轉子或旋轉質量組成，當物體旋轉或傾斜時，轉子會產生相應的旋轉力或角速度。
- 陀螺儀可以用於各種應用，包括導航、飛行控制、運動追踪和姿勢檢測等。



(2) 學習目標

1. 建立一個裝置急墜警報系統，使用「陀螺儀」元件，偵測裝置傾斜和旋轉變化；
2. 展示對「陀螺儀」在 MIT App Inventor 環境裏的應用和量值數據的理解；
3. 展示對程式中使用計時器 (Clock) 元件的了解；
4. 繼續提高對運算思維概念中的「事件」和「分支 / 選擇」的了解；
5. 運用計算思維實踐中的「設計、重用和混合程序 / 編碼」和「測試及除錯」；
6. 建立一個實用的程式與朋友和家人分享，從而發展他們的計算身份認同。

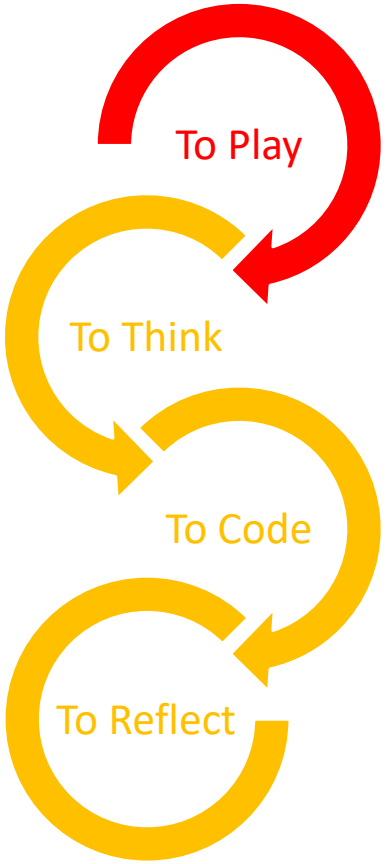
(2) 計算思維主要概念和實踐

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

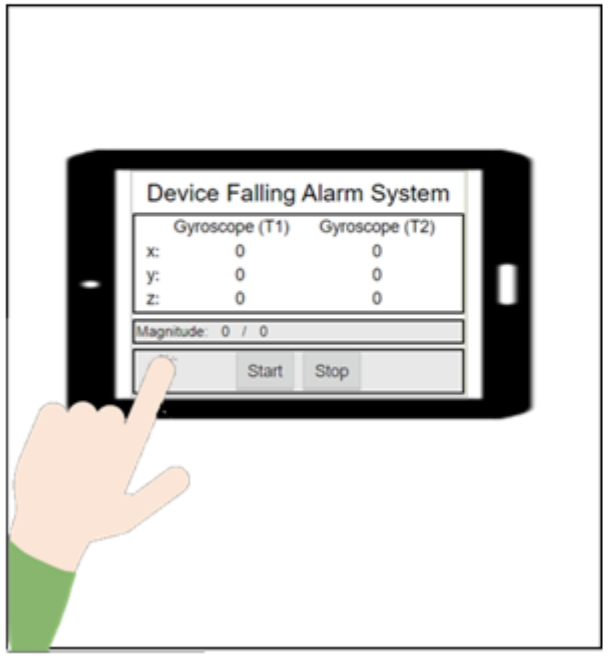
(2) 計算思維態度

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，以構思、創建及改良自己的專案。

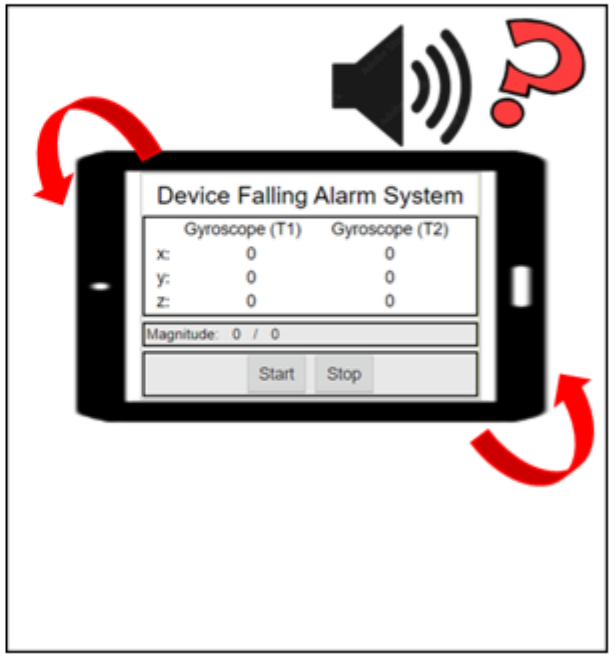
(3) 玩一玩 (To Play)



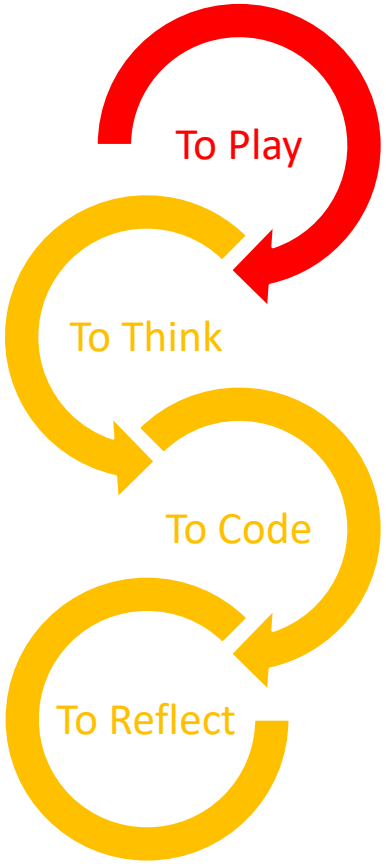
步驟 1：按下 Start（開始）按鈕。稍為移動一下流動裝置，觀察系統界面上的變化。



步驟 2：測試如何能觸發警報系統，並觀察系統界面有甚麼改變。

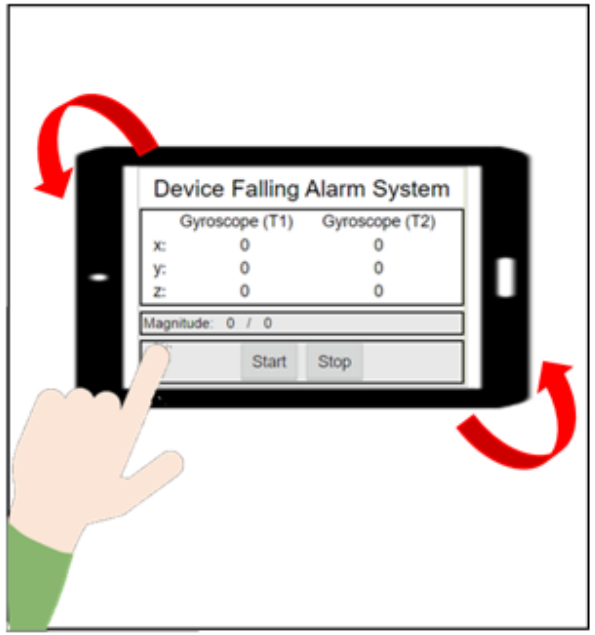


(3) 玩一玩 (To Play)

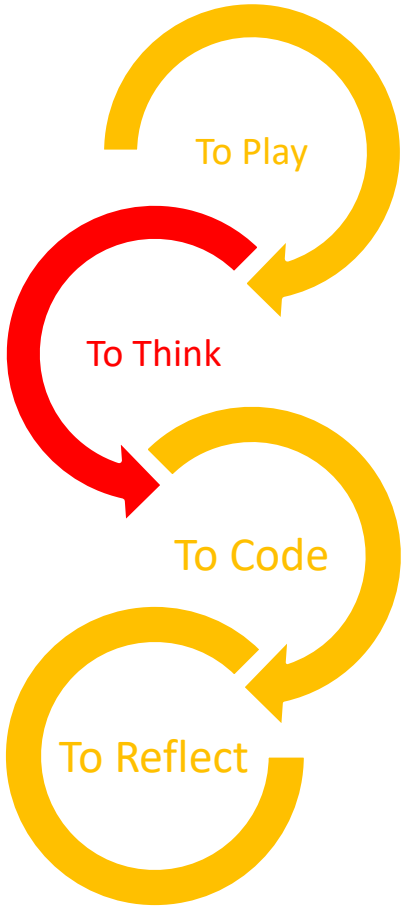


步驟 3：觸發警報後，按下 StopAlert（停止警報）按鈕。觀察系統界面上的變化。

步驟 4：按下 Stop（停止）按鈕。稍為移動一下流動裝置，觀察系統界面上的變化。



(3) 想一想 (To Think)



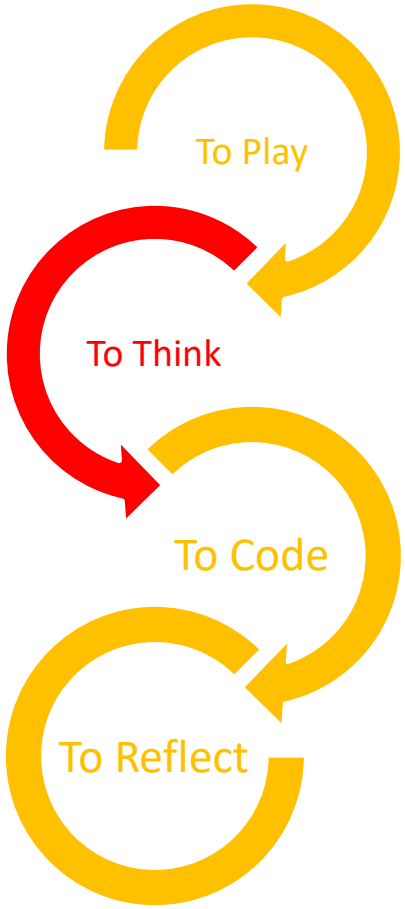
To Think:

- 當連接「裝置急墜警報系統」流動應用程式時，程式介面上的陀螺儀（Gyroscope）的 x-軸、y-軸 和 z-軸及量值（Magnitude）**數值上有沒有變化**？
- 如何**啟動**「裝置急墜警報系統」？

Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3) 想一想 (To Think)



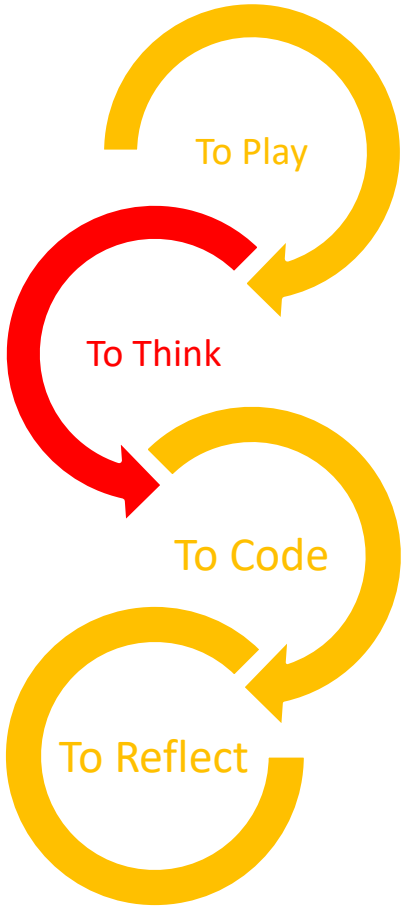
To Think:

- 當連接「裝置急墜警報系統」流動應用程式時，程式介面上的陀螺儀（Gyroscope）的 x-軸、y-軸 和 z-軸及量值（Magnitude）**數值上有沒有變化**？
  由於系統並未開始運作，故此這些數值並未有更新。
- 如何**啟動**「裝置急墜警報系統」？按下 Start 按鈕，系統才開始運作。

Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3) 想一想 (To Think)



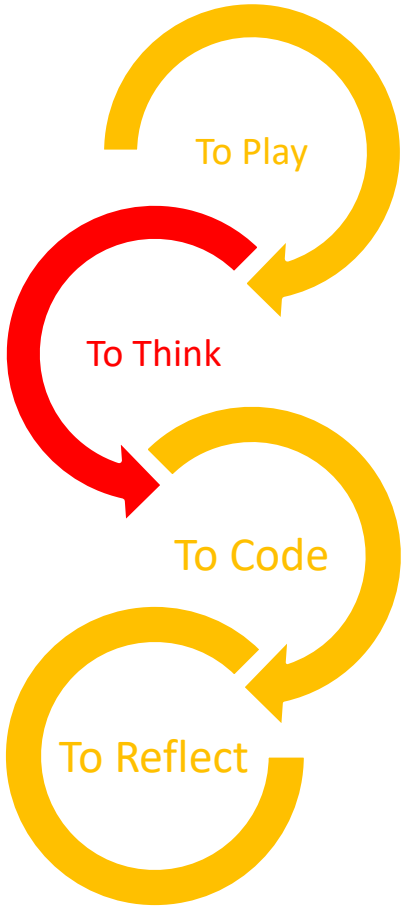
To Think:

- 先將流動裝置放在平穩的桌面，然後慢慢提起、輕輕遙動流動裝置，觀察系統顯示的**數值有甚麼變化**。
- 當**警報觸發**後，系統有甚麼反應？

Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3) 想一想 (To Think)



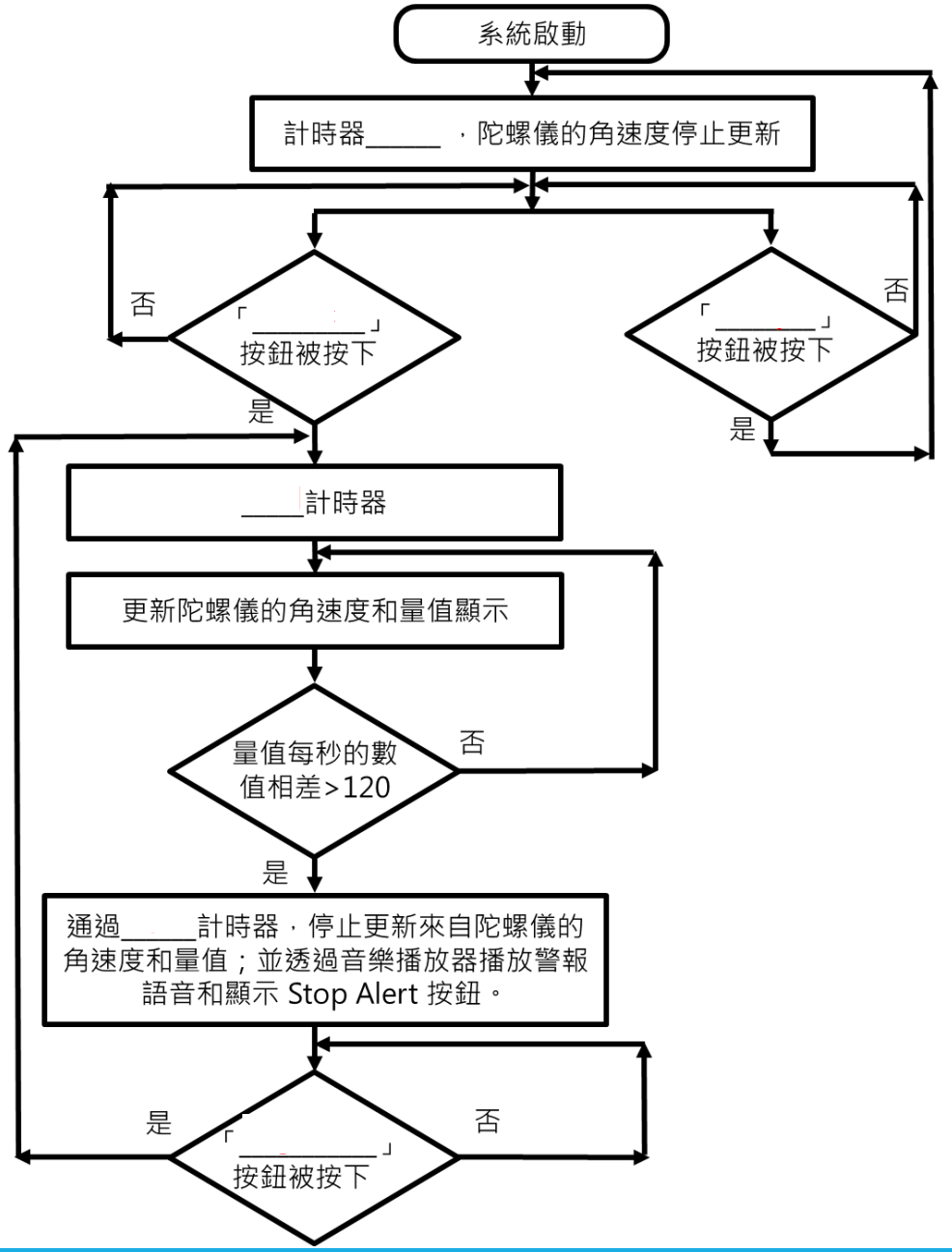
To Think:

- 先將流動裝置放在平穩的桌面，然後慢慢提起、輕輕遙動流動裝置，觀察系統顯示的**數值有甚麼變化**。當流動裝置放在桌面，顯示較小的數值(如低於1)；當輕輕遙動流動裝置，顯示數值會增大。
- 當**警報觸發**後，系統有甚麼反應？系統會停止陀螺儀和量值的數據更新，畫面變成紅色，並播放語音警報和顯示 Stop Alert 按鈕。

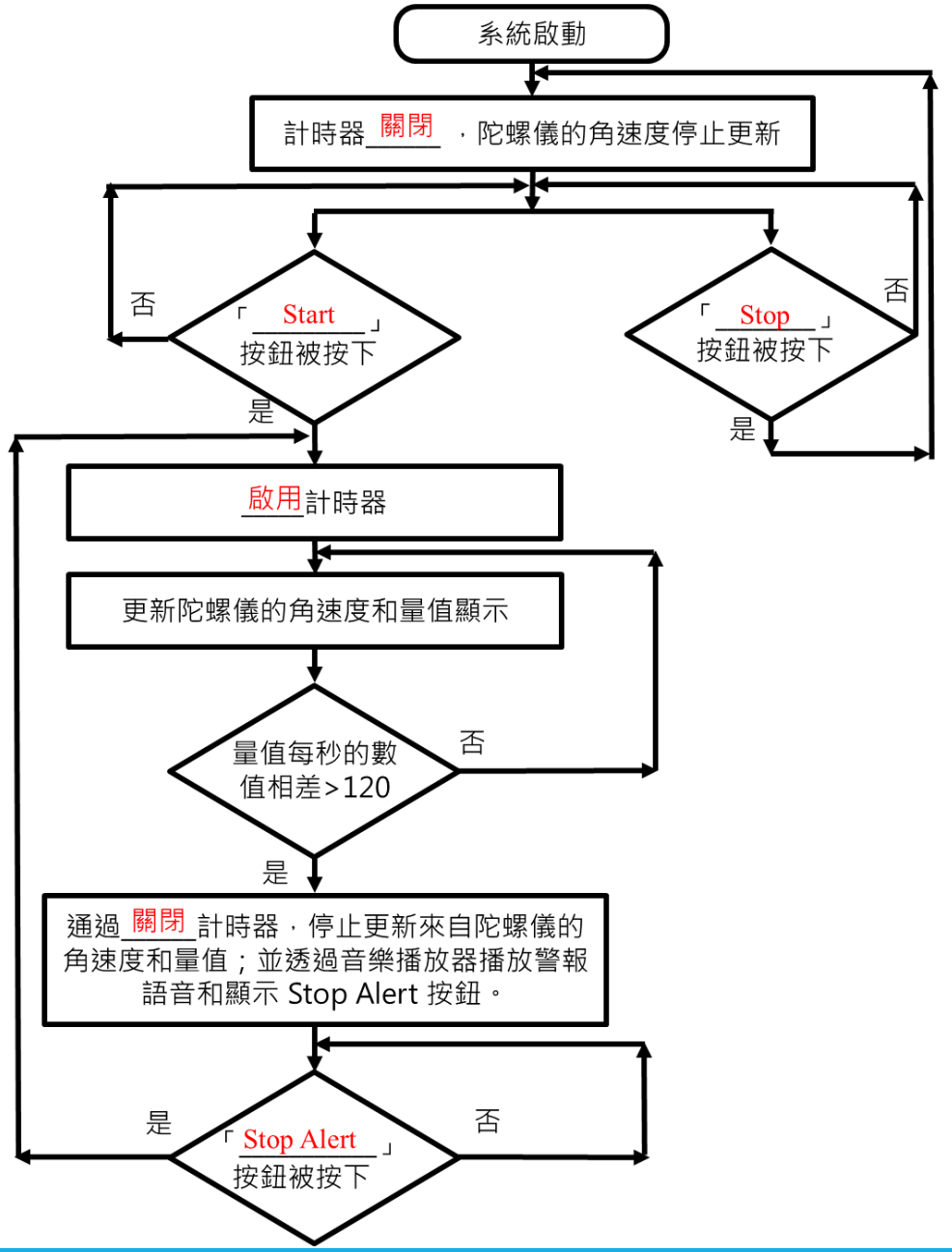
Mentimeter

https://www.menti.com/alkr9rmpmq1c

(3)拆解問題：流程圖


系統啟動
計時器_____，陀螺儀的角速度停止更新
否
「_____」
按鈕被按下
是
否
「_____」
按鈕被按下
是
_____計時器
更新陀螺儀的角速度和量值顯示
量值每秒的數值相差>120
否
是
通過_____計時器，停止更新來自陀螺儀的角速度和量值；並透過音樂播放器播放警報語音和顯示 Stop Alert 按鈕。
是
「_____」
按鈕被按下
否

(3)拆解問題：流程圖


系統啟動
計時器 關閉 ，陀螺儀的角速度停止更新
否
「 Start 」
按鈕被按下
是
否
「 Stop 」
按鈕被按下
是
啟用計時器
更新陀螺儀的角速度和量值顯示
量值每秒的數值相差>120
否
是
通過 關閉 計時器，停止更新來自陀螺儀的角速度和量值；並透過音樂播放器播放警報語音和顯示 Stop Alert 按鈕。
是
「 Stop Alert 」
按鈕被按下
否

(3)拆解問題

我們可以分為兩部分，逐一完成各任務就能製作完整的系統：

1

偵測急墜

1. 獲取並持續更新和顯示陀螺儀的 x-軸、y-軸和 z-軸之角速度數據
2. 計算量值並持續每秒更新一次

2

觸發警報

1. 設置觸發警報的條件
2. 設置觸發警報後的反應：停止陀螺儀更新、播放警報語音、改變背景顏色和顯示「停止警報」按鈕

「裝置急墜警報系統」

(4) 科技教學學科知識 (TPACK)

1. 以範本為基礎，建立專案

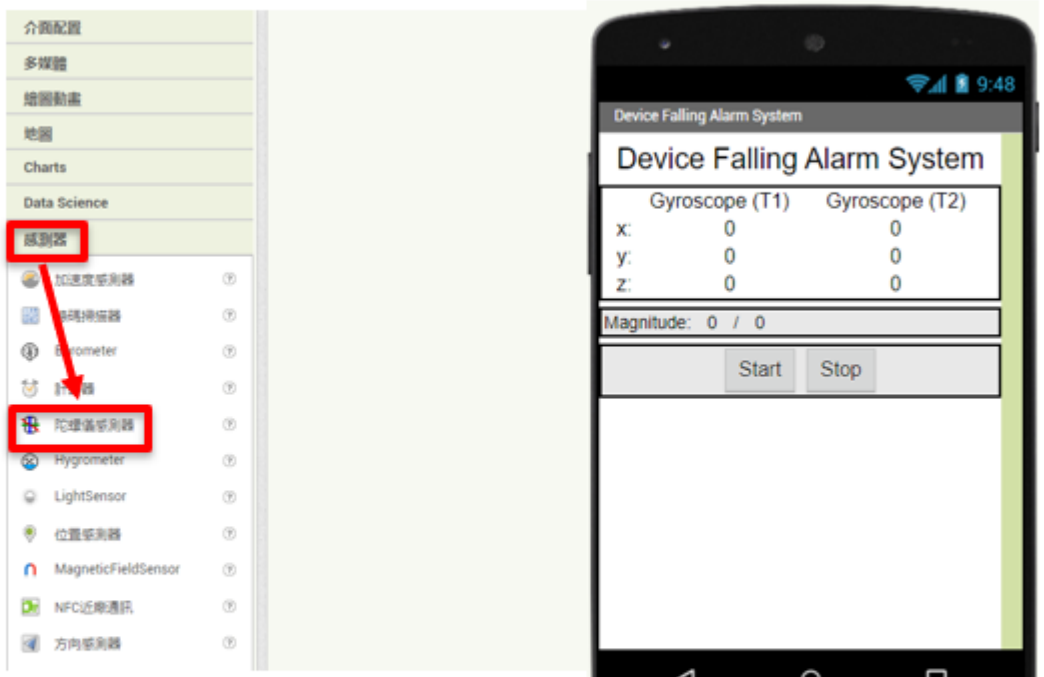
- ☐ 點擊連結下載「裝置急墜警報系統」範本到電腦。
  Device_Falling_Alarm_System_template.aia

- ☐ 匯入範本專案，開始進行編程。

(4) 科技教學學科知識 (TPACK)

2. 從「畫面編排」加入陀螺儀感測器

- 首先在左側項目清單，**感測器 (Sensors)** 中選取**陀螺儀感測器 (GyroscopeSensor)**，並拖到介面中。

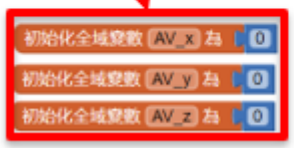


- 成功後，在裝置下方會出現 **陀螺儀感測器 1**

(4) 科技教學學科知識 (TPACK)

3. 從「程式設計」，為顯示陀螺儀感測器數據編程

已經在專案中提供用於存儲 x-軸、y-軸、z-軸的角速度數據的變量。



知識建立：變量

變量在編程用於儲存數值。每個變量都可設置名稱，而它每次只能儲存一個值，此值可以是數字或文字等，亦可在程序中變更新的數值。

知識建立：序列

先將陀螺儀獲取 x-軸、y-軸和 z-軸的角速度數據儲存到變量中，然後再顯示在系統界面中。

(4) 科技教學學科知識 (TPACK)

4. 為顯示量值編程

系統已經成功暫存並在介面顯示角速度數據，我們現在更進一步，利用專案提供的**量值公式**，計算裝置的量值，以判斷長者是否跌倒。

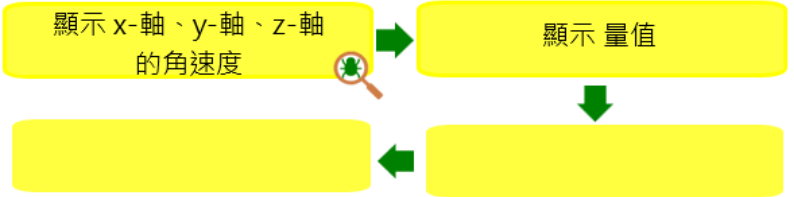
這是當量值大於120，以判斷裝置是否急墜的依據。教師可因應學生能力情況，判斷是否需要提及量值公式。

$$量值公式^{\#} = \sqrt{(\omega x^2 + \omega y^2 + \omega z^2)}$$

測試及除錯

你能成功連接 AI2 Companion 後，界面上能同時顯示現時的量值和上一個量值嗎?

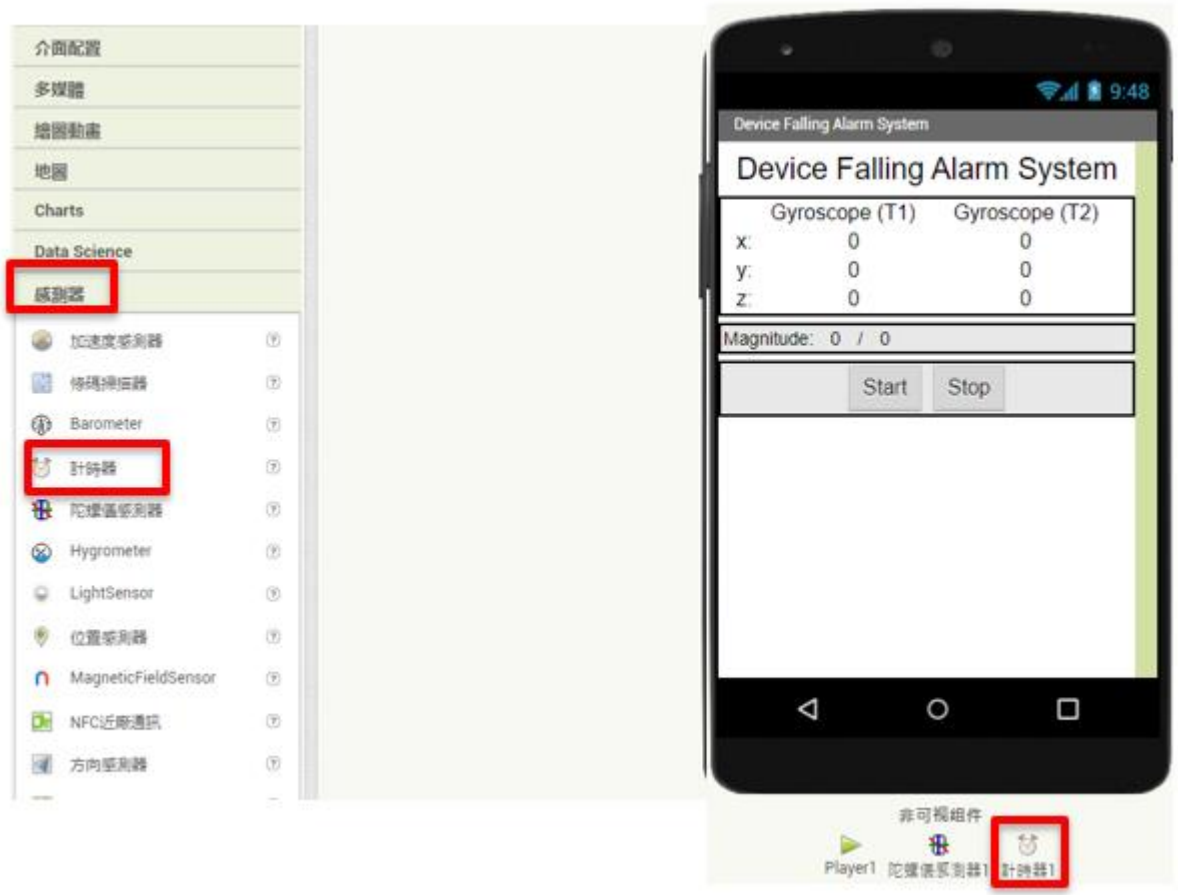
若果未能成功，甚麼地方出錯了？



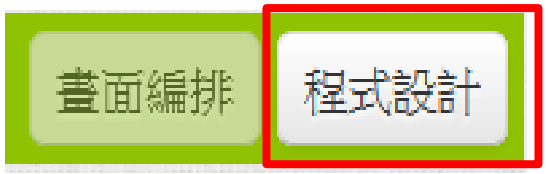
(4) 科技教學學科知識 (TPACK)

畫面編排 程式設計

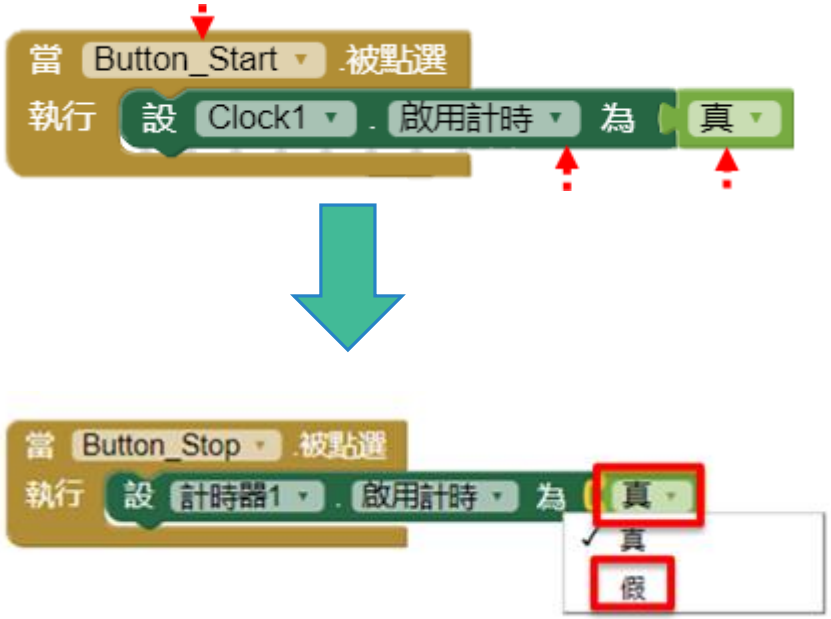
5.從「畫面編排」加入計時器元件



(4) 科技教學學科知識 (TPACK)



6. 從「程式設計」，為計時器按鈕編程



重用及混合Button_Start的指令方塊，到**Button_Stop**

□ 將存儲 x-軸、y-軸和 z-軸的角速度和量值數據的指令方塊拖到**當計時器 1.計時**中。

(4) 科技教學學科知識 (TPACK)

7. 為顯示過去的x-軸、y-軸、z-軸的角速度及量值編程

□ 將存儲過去數據的程序放到**當計時器 1.計時**中。

測試及除錯

你能成功連接 AI2 Companion 後，界面上的數值會因為按下「Start」按鈕而開始每秒更新一次嗎？

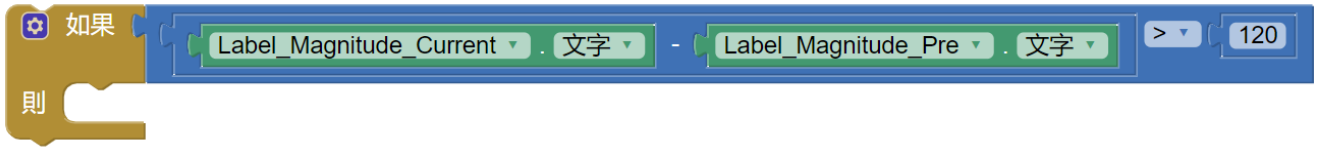
若果未能成功，甚麼地方出錯了？

顯示 x-軸、y-軸、z-軸的角速度 → 顯示 量值

透過計時器控制陀螺儀的數據更新

(4) 科技教學學科知識 (TPACK)

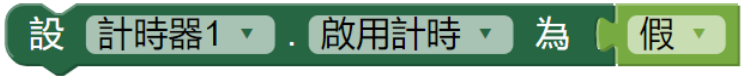
8. 運用條件推理來觸發警報



小提示：陀螺儀的角速度及量值

在「裝置急墜警報系統」中，**陀螺儀的x-軸、y-軸、z-軸的角速度 (Angular Velocity)** 可以幫助我們偵測裝置傾斜和旋轉變化。而我們可透過**量值 (Magnitude)** 公式的每秒數值差是否大於**120**來判斷裝置是否急墜。

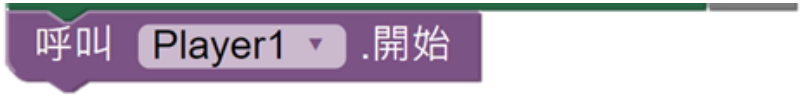
a. 反應（一）：陀螺儀停止介面更新



b. 反應（二）：顯示「停止警報」按鈕

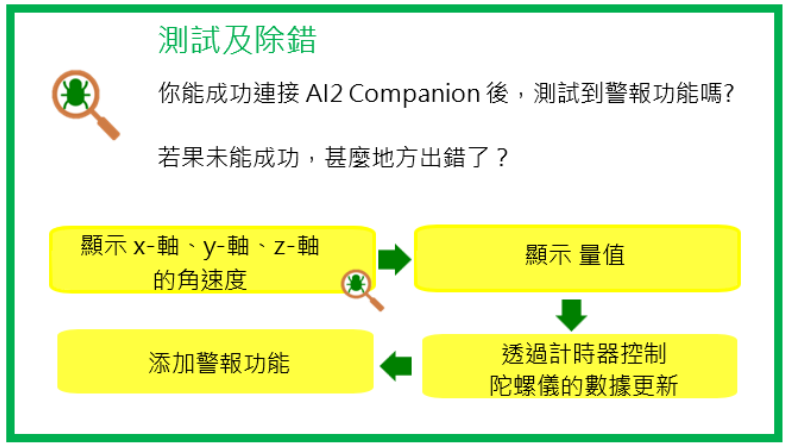
c. 反應（三）：轉換背景顏色

d. 反應（四）：透過音樂播放器播放警報語音



(4) 科技教學學科知識 (TPACK)

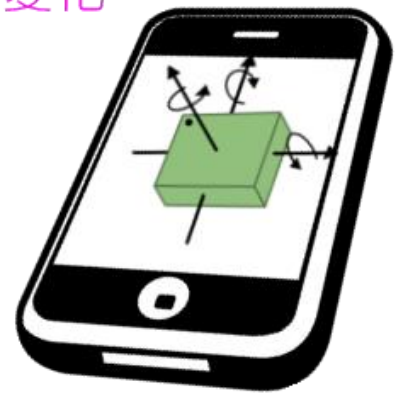
8. 為Stop Alert（停止警報）按鈕編程



(5) 科技內容知識反思，「裝置急墜警報系統」

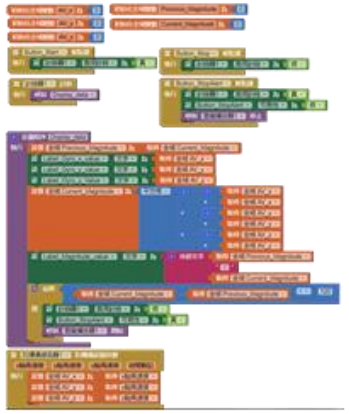
1

裝置上的陀螺儀感測器會持續感測x-軸、y-軸和 z-軸的角速度數值變化。



2

收集角速度數值，進行運算並計算出每秒的量值差。



3

透過每秒的量值差，以判斷用家是否跌倒及需否發出警報。

(5) 科技內容知識反思，啓發數碼創意

提問學生想如何應用「裝置急墜警報系統」到日常生活中？

✓ 例如：如何可以幫助我們自動偵測長者是否跌倒？

當長者跌倒時，我們希望系統能夠做些什麼？

你能想像出一個方法，使系統能夠自動發出求救訊號嗎？

Mentimeter

https://www.menti.com/alkr9rmpmq1c

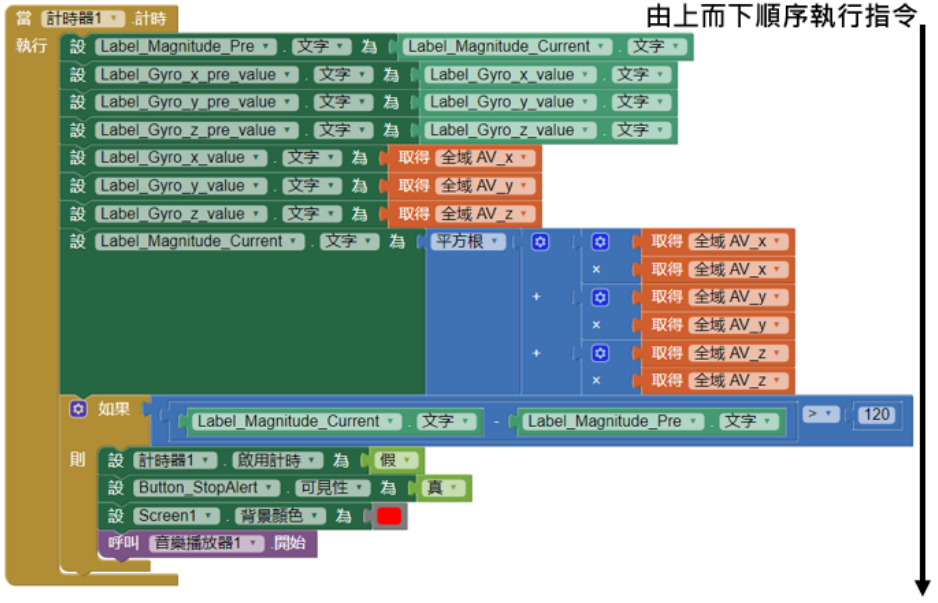
(6) 總結計算思維主要概念和實踐

事件

我們運用了「當 Button_Start/Stop/StopAlert 被點選」、「當計時器 1 計時」、「當陀螺儀感測器 1 陀螺儀狀態改變」事件指令方塊，觸發系統作出反應。

序列

這是編程的重要概念。程序的序列是指執行編程指令的次序。編寫程序時需確定程序指令的執行順序，錯誤的次序會使程序無法正確執行。

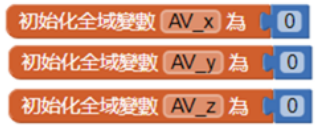


分支 / 選擇

我們在編程運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，它告訴程序當條件為真成立時，「那麼」應該做甚麼。

變量

變量用於儲存數值。它有名稱，每次只能儲存一個數值，但可更新該數值。



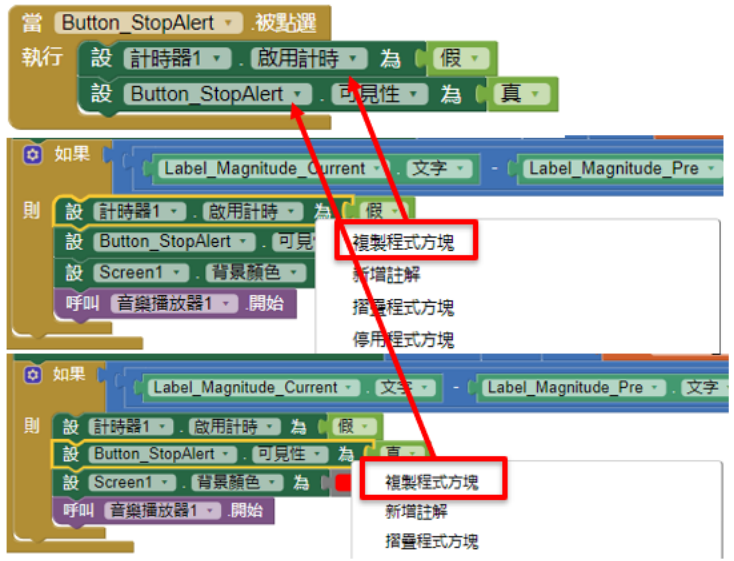
循環

循環是一個重覆的過程並藉此產生一系列的輸出。

(6) 總結計算思維主要概念和實踐

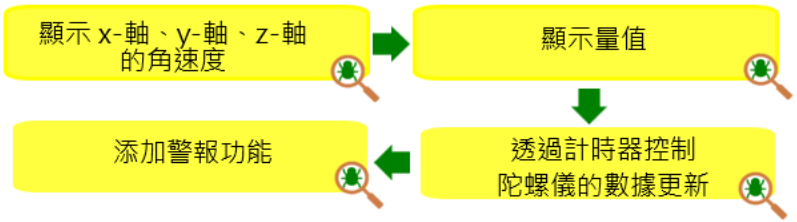
設計、重用和混合程序 / 編碼

我們在編程社群中，運用重用和混合其他編碼十分重要。
例如，我們可以重用和混合一個事件的編碼，應用到其餘的事件。



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。



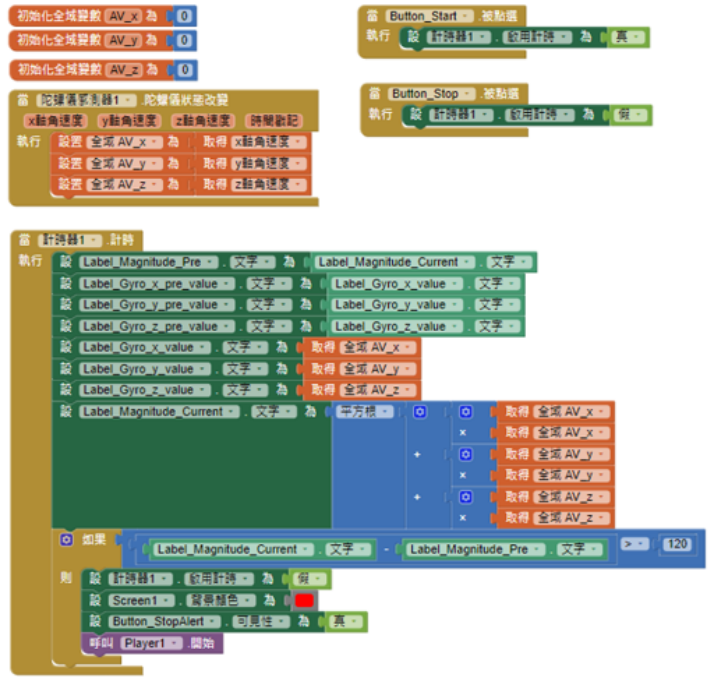
抽象化

抽象化是一個過程，將複雜的問題的描述簡化，方法是隱藏問題中詳細的描述，而保留重要的相關資訊。例如，我們設計一個「裝置急墜警報系統」的應用程式，它會偵測裝置是否正在急墜，如果是的話，就發出警報，就是一個抽象的描述。

算法

算法是一組指示電腦來執行的指令，電腦藉著執行此特定的指令來完成特定的任務。我們的特定任務是偵測裝置是否正在急墜，如果是的話，就發出警報。我們設計一系列指令，讓電腦可以從陀螺儀感測器取得裝置的每秒角速度，並計算量值差，當量值差大於 120 時，系統會判定裝置正在急墜，並觸發警報。

編碼例子：我們根據上述的算法設計來完成程序編寫，以下是「裝置急墜警報系統」的編碼。



(7) 教師自我反思教學法

- 假設每秒量值的數值差超過120，則可能判定裝置正在急墜，並觸發警報。您覺得這樣的設定合適嗎？
  例如：合適，因為當流動裝置下墜時，能觸發警報。
  例如：不合適，因為手持流動裝置時，稍快速的移動都會觸發警報。

- 你能否想到其他延伸的挑戰任務?

- 同學有否被啟發去運用及整合學科知識去製作屬於他們的程式?

單元示例：專題習作 Final Project

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

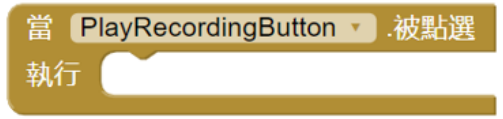
學習目標

1. 學生能定義一個待解決的問題，並提出建議以運用App Inventor編程解決該問題；
2. 設計一個App Inventor專案來解決定義的問題；
3. 應用計算思維概念和實踐來完成App Inventor專案；
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。

重溫主要元件

按鈕

使用者可透過點選、長按、放開按鈕等（事件）來觸發程序執行一些反應動作。使用者也可以改變按鈕的外觀，如顏色、大小、形狀等。

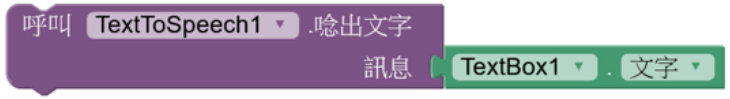


文字語音轉換器

運用 「文字語音轉換器」(**TextToSpeech**) 元件能夠使程式讀出任何你提供的字串。

你可以把想要程式讀出的文本放進文字（text）指令方塊中。

你的程式還可以讀出「文字輸入盒」(textbox) 的內容，或其他使用者界面元件的內容。

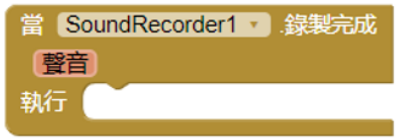


音樂播放器

用於播放音效的多媒體元件，是一個**「不可見的元件」**。在「畫面編排」及「程式設計」視窗中，可用「來源」屬性來指定音效來源，適合於播放較長的音效檔，如歌曲。

錄音機

用於錄製聲音的多媒體元件，是一個**「不可見的元件」**；運用流動裝置的麥克風，能同時收錄流動裝置揚聲音播放的聲音及環境聲音。



水平配置

「水平配置」元件是用於「畫面編排」，可讓放進「水平配置」中的元件水平排列。如果希望元件上而下排列的話，則可用「垂直配置」元件。

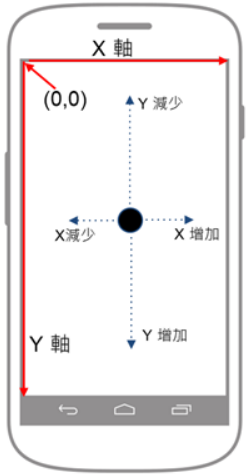
重溫主要元件

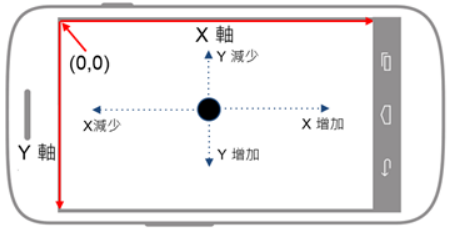
畫布及圖像精靈

你必需加入畫布（Canvas）這個「繪圖動畫」類型的元件至「工作面板」，然後加入球形精靈（Ball）或圖像精靈（ImageSprite）到畫布。

原點（0,0）是位於畫布的左上角。當精靈的 X 座標值增加時，它會向畫面的右邊移動，當精靈的 Y 座標值增加時，它會向畫面的下邊移動。

就好像座標一樣，你可以透過設定球形精靈或圖像精靈的 X 及 Y 座標值定位，來設定它們在畫布上的位置。





圖像精靈被碰撞

當精靈跟另一圖像精靈碰撞，便觸發「碰撞」事件，可以運用 **當....碰撞** 指令方塊，執行碰撞後被觸發的程序。

這個指令方塊可以用作測試一個球形精靈有否跟圖像精靈碰撞。當球形精靈（如 **Ball1**）跟圖像精靈（如 **GoldSprite**）碰撞後，被觸發的程序可以放於 **則** 內。

球形精靈 / 圖像精靈到達邊界

如果你希望精靈到達邊界時發出指令（執行被觸發的程序），可運用 **當（球形精靈 / 圖像精靈）.到達邊界** 指令方塊。

重溫主要元件

切換「畫面」

在 App Inventor 中「畫面」與「螢幕」意思相似。如果要從一個「畫面」切換到另一個「畫面」，你可以用 **開啟另一畫面** 指令方塊，並配合「畫面名稱」（或內容為「畫面名稱」的文字方塊）。在下列例子中，按 **SwitchButton** 按鈕時，會觸發 **當 SwitchButton.被點選** 的「事件」指令方塊，開啟 **LearningScreen**「畫面」。

你也可以透過 **開啟其他畫面並傳值** 指令方塊，在「畫面」之間傳遞信息。在下列例子中，當你長按 **CatStreet** 按鈕，便會打開另一個名為 **LocationScreen** 的「畫面」，同時傳送初始值 **"Cat Street"** 。

地圖

以下指令方塊可使畫面 **Screen1** 啟動時，將地圖中心平移到指定的緯度和經度，然後將縮放級別按縮放數值調整。

照相機

以下指令方塊為例，**呼叫 Camera1.拍照** 指令方塊呼叫照相機進行拍照；而當拍照完成後，會自動觸發 **當 Camera1.拍攝完成** 的「事件」指令方塊，**執行** 指定的程序。

重溫主要元件

微型資料庫

微型資料庫（TinyDB）則應用程式提供了一種永久性的資料儲存方式。

呼叫（微型資料庫）.取得數值 的指令方塊能檢索儲存在指定標籤下的值。如果沒有這樣的標籤，則傳回 valueIfTagNotThere。

呼叫（微型資料庫）.儲存數值 的指令方塊能將儲存值（valueToStore）儲存在指定標籤下。即使應用程式重新啟動時，微型資料庫的儲存空間仍會保留在流動裝置（如手機）上。

計時器

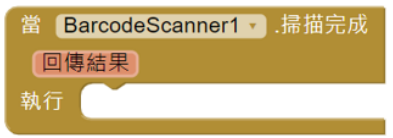
運用裝置內部時鐘提供即時時間。計時器可以設定時間間隔來觸發事件；計時器亦可執行時間計算、操作和轉換。

以下指令方塊為例，當每隔一段時間（由「計時間隔」的值決定），會自動觸發「**當（計時器 1）.計時**」指令方塊，**執行** 指定的程序。

條碼掃描器

用於讀取條碼訊息的感測器元件，是一個「**不可見的元件**」；它運用流動裝置的相機來掃描條碼。

掃描完成後，將自動觸發 **當....掃描完成** 事件，我們可運用 **取得「回傳結果」** 指令方塊獲得條碼掃描器讀出的（文字）結果。



陀螺儀感測器

提供來裝置內部的陀螺儀感測器數據。

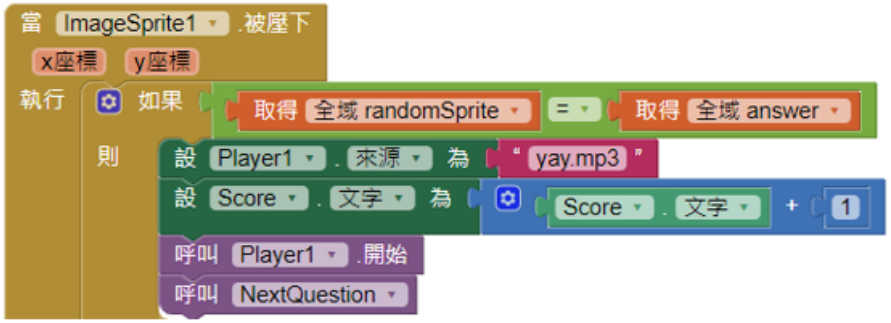
以下指令方塊為例，當 **陀螺儀感測器 1** 的讀數有改變，會自動觸發「**當（當 陀螺儀感測器 1）.陀螺儀狀態改變**」指令方塊，**執行** 指定的程序。

重溫計算思維主要概念和實踐

事件

我們期望應用程式能按使用者的操作（如點選按鈕、壓下圖像精靈，這些操作被視為「事件」）作出反應。我們運用「事件」指令方塊，將「事件」扣連到應用程式需要作出反應而執行的程序（被觸發的程序）。

例如：「當 **ImageSprite1**.被壓下」（事件），就會檢查答案是否正確（被觸發的程序）。



序列

這是編程的重要概念。程序的序列是指執行編程指令的次序，留意一些指令必須有其特定次序。編寫程序時需確定程序指令的執行順序，錯誤的次序或會使程序無法正確執行。

例如，必需先設置 **randomSprite** 的值，然後才能設定 **ImageSprite** 的圖片，否則會無法正確配對數字球的圖片。

分支 / 選擇

我們在編程使用條件句進行推理，讓電腦做決定。條件句總有「如果」的部分，它告訴程序當條件為真成立時，「則」應該做甚麼。

例如：[條件句]「如果」答案正確

[結果] 「則」播放歡呼聲、加 1 分，以及轉去下一題

循環

循環是一個重覆的過程並藉此產生一系列的輸出。

重溫計算思維主要概念和實踐

變量

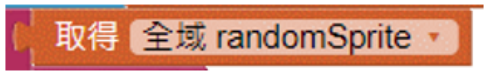
在 App Inventor 中，變量的意思與「變數」相同。變量用於儲存數值（包括數字、文字、布林邏輯真假值等）。

變量有名稱，每次只能儲存一個數值，我們可在指令程式在運作過程中更新該數值。

使用「**初始化全域變數**（變量名稱）**為...**」指令方塊能按我們給予的命名建立變量。以下指令方塊為例，建立 **randomSprite** 變量，並將其數值設為 0（數字）。

使用「**設置**(變量)**為...**」指令方塊能更改變量的值。以下指令方塊為例，將 **randomSprite** 變量設定 / 更新為清單變量 **randomAnswer** 中隨機抽出的一個項目。

使用「**取得**(變量)」指令方塊能取得變量現時的值。



清單

清單是一種數據結構，包含「索引」和「清單項目」;「數據結構」是電腦科學的名詞，代表儲存多項資料於一個變量中。

以下指令方塊為例，將 **numberList** 清單變量設定 / 更新為有三個項目，每個項目均為 1 至 9 之間的隨機整數。

使用「**選擇清單...中的索引值為...的清單項目**」指令方塊能抽取清單變量中的某一個項目。

以下指令方塊為例，會傳回 **numberList** 清單變量中的第一個項目（索引值為 1）。

使用「**求取清單的長度 清單...**」指令方塊能得知清單內有多少個項目（清單的項目數量）。

使用「**清單是否為空？ 清單...**」指令方塊能得知清單是否空白；當清單是空白時，方塊的值為「真」; 如清單包含元素，方塊的值為「假」。

重溫計算思維主要概念和實踐

設計、重用和混合程序 / 編碼

運用重用和混合其他編碼對編程十分重要。

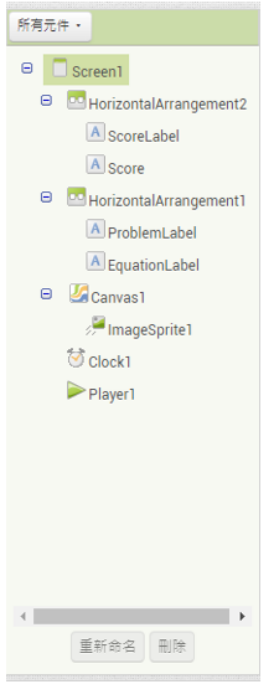
例如，我們可以重用和混合一個事件的編碼，應用到其餘的事件。

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。

命名

為元件命名有助識別其功能及進行區別。



重溫基本程序編寫結構的概念

抽象化

抽象化是一個過程，將複雜的問題的描述簡化，方法是隱藏問題中詳細的描述，而保留重要的相關資訊。

例如，我們想設計一個數學遊戲程式，使用者要根據題目，找出正確答案的數字球，就是一個關於數學遊戲的抽象描述。

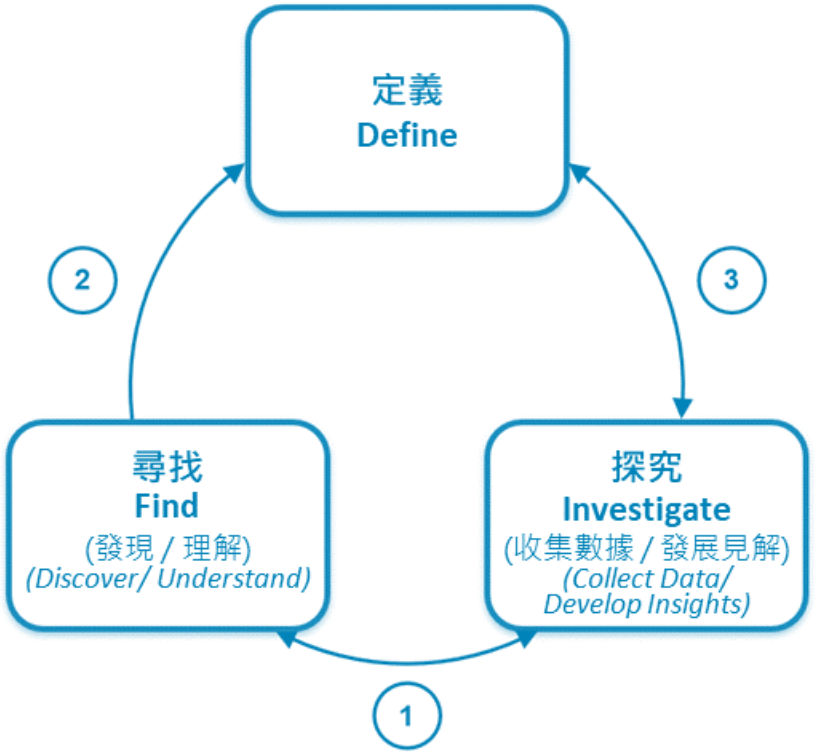
算法

算法是一組指示電腦來執行的指令，電腦藉著執行此特定的指令來完成特定的任務。例如，特定的任務是製作三個數字加法運算的遊戲程式，數學問題類似 "5 + 6 + 5 = ?" 的形式，問題的數字將隨機生成，以確保每次遊戲都是新挑戰；使用者需要點擊有正確答案的數字球；如果答案正確，遊戲會有歡呼音效，否則會有沮喪音效；另外，答案正確時會增加 1 分，反之會扣減 1 分。

編碼例子：我們根據上述的算法設計來完成程序編寫，以下是答案正確或錯誤時所發生的事件編碼。

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/MakeCode+micro:bit units**
  重溫並參考所學單元知識



(Kong, 2022)

同創作 (To create)

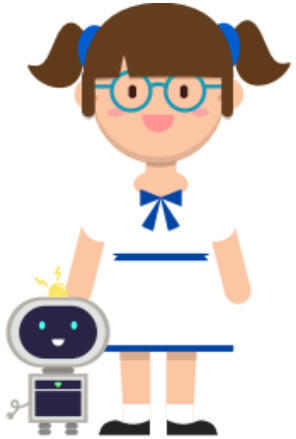
同創作 (To create)：透過修改專案並加入自己的構思，學生對跟他人分享自己製作的專案而感到自豪。

定義問題

1. 你身邊有沒有人遇到一些問題？ 她 / 他是誰？

小學低年級生。

我想為同學設計一個遊戲。



2. 他 / 她面臨什麼問題？

我們發現一些同學在學習數學上遇到了挑戰，特別是在學習加法時。

3.你希望如何利用 App Inventor來幫助他 / 她去解決這個問題？ 請描述你的想法。

圈出專案的類型：遊戲 / 實用流動應用程式

主題名稱：「數學遊戲」教育程式

程式目的：專為小學低年級生提供有趣且有效的方式學習加法。這個程式將通過互動的方式，包含音樂、圖像、動畫，使他們能夠更輕鬆地理解及有效掌握加法的概念。

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

配合音樂科樂器介紹的流動應用程式，讓學生創造有特式的電子樂器

貼近學生的學習的流動應用程式，例如配合中文科或英文科讓學生編寫針對某單元詞彙的學習程式，幫助學生溫習

由「撞球遊戲」發展成「彈珠遊戲」，可提升學生的創意及設計遊戲規則的能力

將「普通話聽力測驗示範」改為輔助學生準備中 / 英文默書的流動應用程式

建立在課堂使用的「抽籤」流動應用程式（代替上課 / 活動時人手抽籤），並加入記錄課堂表現分數的功能（建立「學生表現龍虎榜」）

學校地圖（介紹學校各樓層予小一學生或到訪嘉賓）

例子一：

「數學遊戲」教育程式

例子一：「數學遊戲」教育程式 (範例供教師參考)

在這單元，你會設計並製作一個有關數學的教育遊戲程式，讓使用者學習三個數的加法運算。

1. 描述一下已定義的教育遊戲程式的功能。

製作一個三個數字加法運算的遊戲：數學問題類似 "5 + 6 + 5 = ?" 的形式，問題的數字將隨機生成，以確保每次遊戲都是新挑戰。介面設計能吸引初小學生，使用者需要點擊有正確答案的圖像精靈。答案正確時，遊戲會有歡呼音效，否則會有沮喪音效；另外，答案正確時會增加 1 分，反之會扣減 1 分。

例子一：「數學遊戲」教育程式 (範例供教師參考)

「畫面編排」(Designer View):

2. 選出程式需要用的元件，在方格內加上 "✓"及寫出其功能：

元件（選出最少兩個）	
✓ 音樂播放器	✓ 標籤
☐ 按鈕	☐ 圖像
☐ 文字輸入盒	✓ 水平 / 垂直配置
☐ 條碼掃瞄器	☐ 標記
☐ 陀螺儀	✓ 其他：畫布 _____
☐ 文字語音轉換器	_____
✓ 圖像精靈	_____
✓ 計時器	

例子一：「數學遊戲」教育程式 (範例供教師參考)

3. 畫面快照 (Screenshots)

初步構思並畫上程式屏幕的設計畫面，並寫出元件名稱。

Screen1

4. 元件清單

列出你的程式中所需的所有元件。

小提示
程式亦可以有多個畫面 (Screens)

畫面 1		
元件種類（按鈕、文字輸入盒等）	**名稱**	**特別屬性（字元、顏色、對齊等）**
音樂播放器	Player1	來源 1：yay.mp3 來源 2：sad-trombone.wav 音量：50
圖像精靈	ImageSprite1	高度 / 寬度：75 像素 可見性：真 圖片：全域 randomSprite.png
計時器	Clock1	持續計時：真 啟用計時：真 計時間隔：1000
標籤	ScoreLabel	Score: 0
標籤	ProblemLabel	? + ? + ? = ?

描述你的元件如何運作。例如，當按動 **Button1** 時，就.....

ProblemLabel：顯示數學問題，類似 "5 +6 + 5 = ?" 的形式。

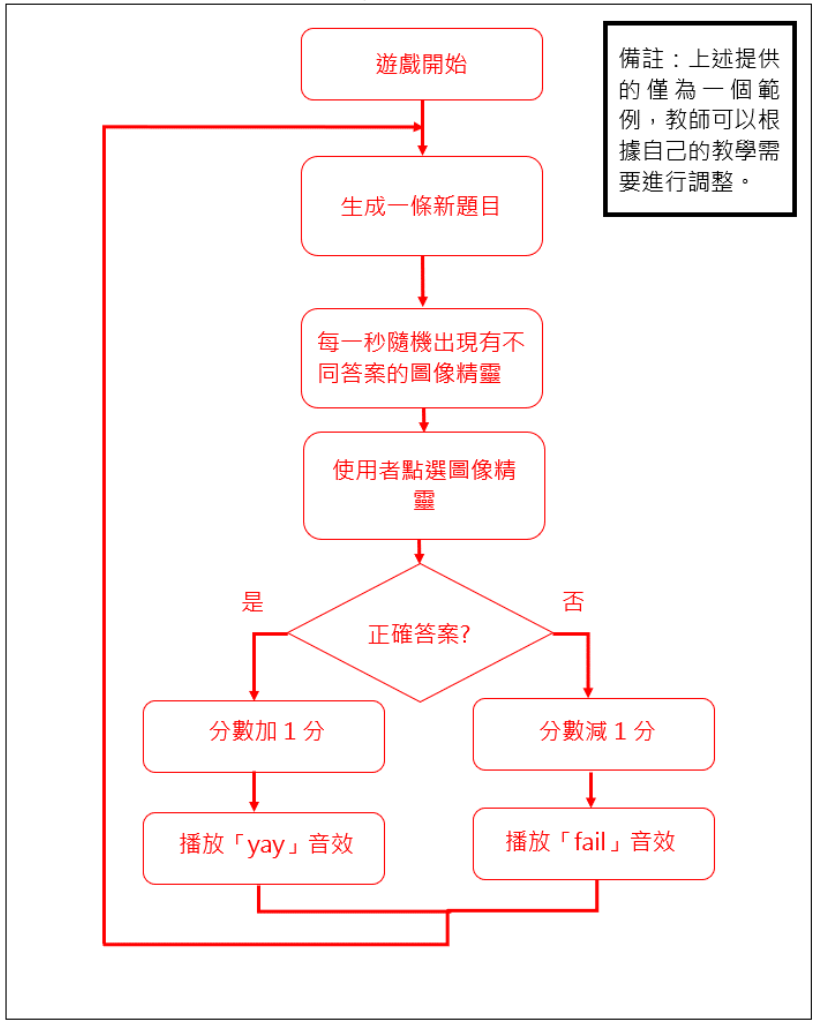
Clock1：每一秒（1000ms）隨機出現有不同答案的圖像精靈

Player1：當點擊正確答案時，會有歡呼音效，否則會有沮喪音效。

ScoreLabel：答案正確時會增加 1 分，反之會扣減 1 分。

例子一：「數學遊戲」教育程式 (範例供教師參考)

5. 運用流程圖等工具，把你的想法建構出來。



「程式設計」(Blocks Editor)：

6. 參考上頁的元件清單及流程圖，**前往程式設計頁面開始編程** 。

「內建方塊欄目」(Built-In Drawer)：使用者可在此找出程式基本行為的指令方塊，如 App Inventor 內建的控制和邏輯指令；**「特定元件的方塊欄目」(Component-Specific Drawers)**：使用者可在此找出特定元件（如由使用者加入的按鈕和標籤）的相應的指令方塊，包括事件、元件屬性。

例子一（甲）評級：優異

簡評

畫面編排: 專案能美觀地安排使用者界面，加上富有趣味的互動元件，如計分標籤、在隨機位置出現的數字球、以及答題音效，讓使用者從開始到結束都能投入其中。

程式設計: 專案能具體地定義設計遊戲的規則，例如考慮到隨機出現的數字球有 3 個可能性。同時能利用程序將複雜的任務分解，以及有效地運用計算思維概念來簡化和優化程序，高效及有系統地完成所有任務。

例子一（乙）評級：良好

簡評

畫面編排: 有邏輯和美觀地安排使用者界面，並且具備聲效及圖像，大致能照顧使用者的需要。如果能加入玩法的指示，或者計時器倒數的提示，相信能讓使用者有更優質的體驗。

程式設計: 專案能制定合適的計劃來解決問題，例如使用者需要在 3 秒內選擇正確答案。不過，專案並未考慮到選項一及二的答案有機會重覆。另外，當使用者重覆遊玩時，會發現選項一永遠顯示正確答案，降低了遊戲的趣味性。

例子一（丙）評級：繼續努力

簡評:

畫面編排: 使用者界面較為基本，只有簡單的文字顯示及輸入功能，未能有效吸引使用者。另外，不少操作或介面設計不清晰，例如在甚麼地方輸入答案？如何知道所輸入答案的是否正確？

程式設計: 專案能制定簡單計劃來解決當中的小部分問題，例如設計簡單的答題練習程式。不過，似乎未能達到當初的預期效果，要製作一個加法運算的遊戲，來吸引初小學生學習數學。

自行創作

1. 描述一下已定義的程式功能。

_____

_____

_____

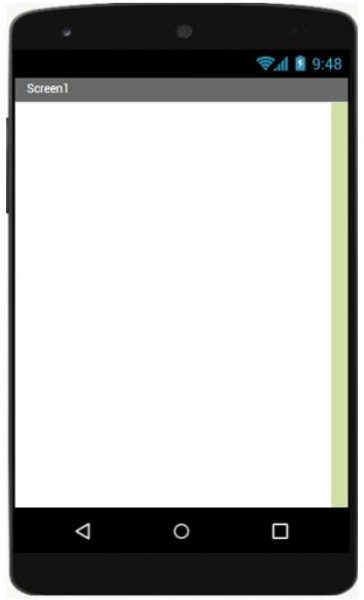
_____

「畫面編排」(Designer View):

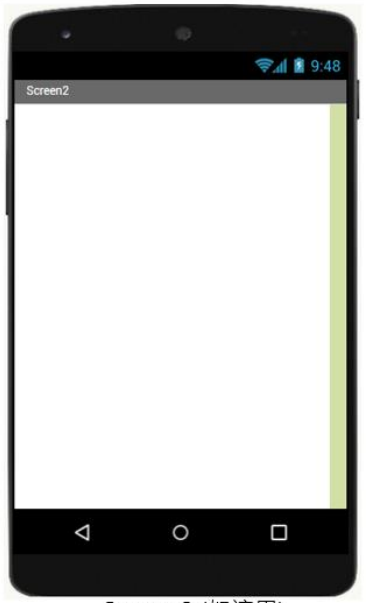
2. 選出程式需要用的元件，在方格內加上 "✓"及寫出其功能：

元件 （選出最少兩個）	
☐ 音樂播放器	☐ 標籤
☐ 按鈕	☐ 圖像
☐ 文字輸入盒	☐ 水平 / 垂直配置
☐ 條碼掃瞄器	☐ 標記
☐ 陀螺儀	☐ 其他：_____
☐ 文字語音轉換器	_____
☐ 圖像精靈	_____
☐ 計時器	

3. 畫面快照 (Screenshots)
初步構思並畫上程式屏幕的設計畫面，並寫出元件名稱。



Screen1



Screen2 (如適用)

自行創作

4. 元件清單

列出你的程式中所需的所有元件。

畫面 1		
元件種類（按鈕、文字輸入盒等）	**名稱**	**特別屬性（字元、顏色、對齊等）**

描述你的元件如何運作。例如，當按動 **Button1** 時，就.....

畫面 2		
元件種類（按鈕、文字輸入盒等）	**名稱**	**特別屬性（字元、顏色、對齊等）**

描述你的元件如何運作。例如，當按動 **Button1** 時，就.....

自行創作

5. **運用流程圖等工具**，把你的想法建構出來。

「程式設計」(Blocks Editor)：

6. 參考上頁的元件清單及流程圖，**前往程式設計頁面開始編程** 。

「內建方塊欄目」(Built-In Drawer)：使用者可在此找出程式基本行為的指令方塊，如 App Inventor 內建的控制和邏輯指令；**「特定元件的方塊欄目」(Component-Specific Drawers)**：使用者可在此找出特定元件（如由使用者加入的按鈕和標籤）的相應的指令方塊，包括事件、元件屬性。

反思初步設計

這些問題可以引導學生反思他們的初步設計，從問題定義到算法設計和運用App Inventor元件等。
詳細的評分準則，教師可參考文件：專題習作評分標準（小組 / 個人）。

問題定義：
我們是否清晰地辨識和定義我們試圖解決的問題？
我們是否能夠以他人能理解的方式表達問題？並提出有效的的解決方案 / 設計？

算法設計：
我們的程序是否按照合乎邏輯的步驟（序列）進行？
我們的程序是否以有效的方式設計，命名清晰？
我們在程序是否有效運用循環、分支 / 選擇和變量？
我們是否有機會進一步簡化或優化我們的程序？例如加入清單或程序？

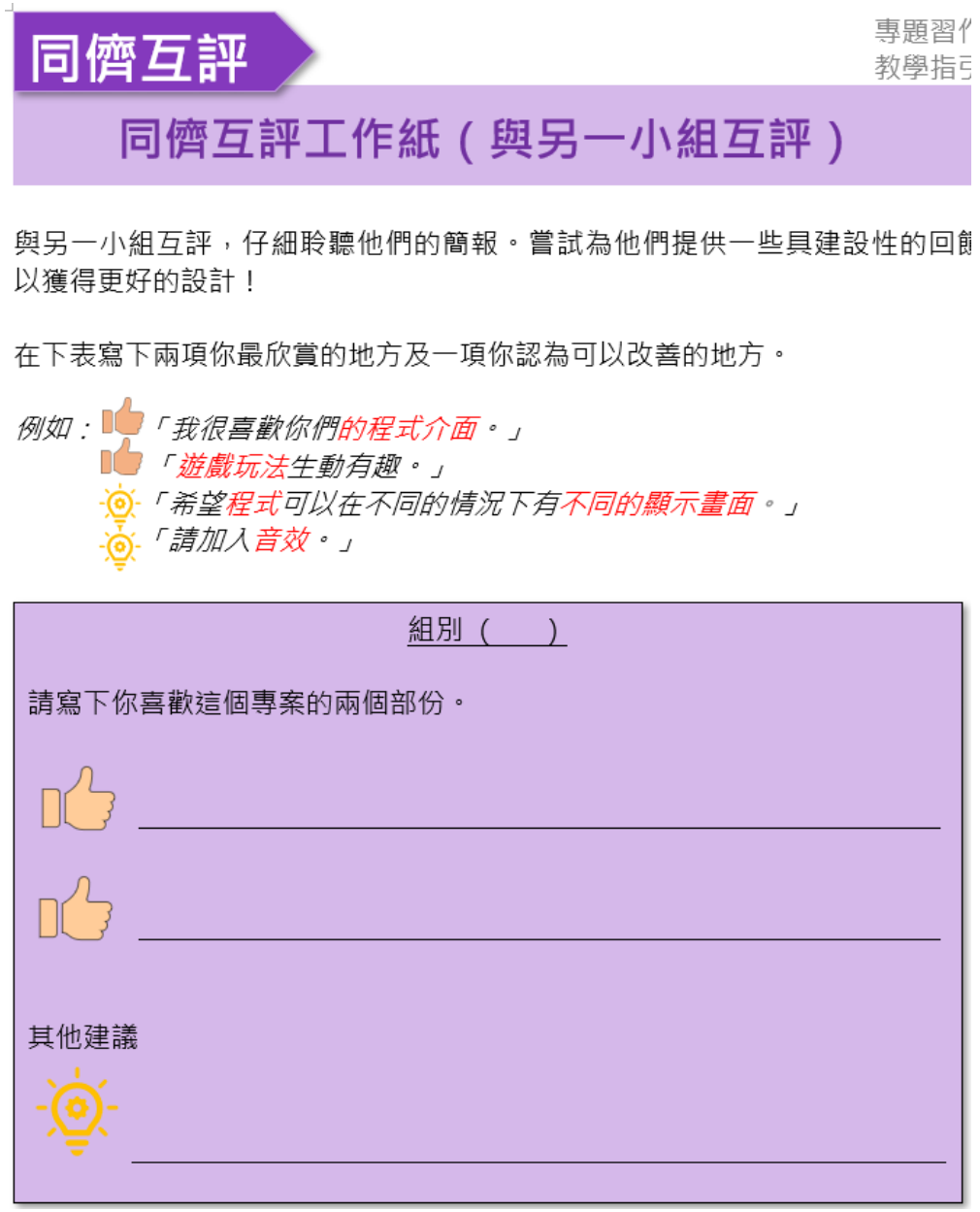
運用App Inventor元件：
我們在App Inventor中的元件和指令方塊中是否充分發揮了作用？

創意和美學，以及使用者的互動體驗：
使用者界面有沒有運用水平布局和垂直布局，並有邏輯和美觀地安排元件？
我們的程序是否能有效創建出互動豐富和能讓使用者積極參與的體驗？例如，我們是否注意到視覺細節？我們的設計如何能進一步增強使用者體驗？
我們如何運用不同的感官元素來吸引使用者？

專題習作待辦清單

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	定義問題	於第一教節完成
第二教節	畫面設計	於第二教節完成

同儕互評



同儕互評

專題習作
教學指引

同儕互評工作紙（與另一小組互評）

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

在下表寫下兩項你最欣賞的地方及一項你認為可以改善的地方。

例如：「我很喜歡你們的程式介面。」
「遊戲玩法生動有趣。」
「希望程式可以在不同的情況下有不同的顯示畫面。」
「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

_____

小組簡報

小組簡報

按照教師的指示，你可以先透過「同儕互評工作紙（與另一小組互評）」改善你的專案，再進行小組簡報。

當進行小組簡報和示範時，可以：

1. 介紹自己及組員：

   _____

2. 簡介專案的主題、目的、所設計的方案如何有助解決問題：

   _____

   _____

   _____

3. 在製作專案最成功的部分：

   _____

4. 在製作專案中克服了哪些困難：

   _____

5. 專案有甚麼地方可以改良：

   _____

自我評估

齊反思

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，你扮演了甚麼角色？
   ☐組長 ☐編程人員 ☐平面設計師 ☐其他_____

4. 簡介你完成的工作？_____

   _____

5. 你覺得你是個好的組員嗎？你在專題研習中有沒有互相幫助，並且尊重和鼓勵你的組員嗎？ ☐是 / 有 ☐我能做得更好

   因為我……_____

   _____

專題習作評分標準 (小組)

範疇	優異	良好	尚可	繼續努力
定義問題（16%）	能透過仔細的觀察和具體的分析來清楚地辨識和定義專題習作需解決的問題，從而提出適切可行及有效的解決方案 / 設計，並制定周詳計劃來解決該問題。 分數：13 – 16	大致能透過觀察和分析來辨識和定義專題習作需解決的問題，且提出可行的解決方案 / 設計，並制定合適的計劃來解決問題。 分數：9 - 12	嘗試辨識和定義專題習作需解決的問題，且提出一些可行的解決方案 / 設計，並制定簡單計劃來解決當中的小部分問題。 分數：5 - 8	未能辨識和定義專題習作需解決的問題，提出的解決方案 / 設計只能解決當中的個別問題。 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，命名清晰，能出色地運用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，甚至加入清單（list）或程序（procedure），從而進一步簡化和優化程序，高效及有系統地完成所有任務。 分數：31 - 40	程序設計中大致能呈現合乎邏輯的序列（sequence）。程序以有效的方式設計，命名清晰，多能善用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，從而簡化和優化程序，完成大部分任務。 分數：21 - 30	程序設計中部分能呈現合乎邏輯的序列（sequence），大致能恰當運用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，完成小部分任務。 分數：11 - 20	程序設計中較少能呈現合乎邏輯的序列，程序中少能正確運用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，任務大多未能完成。 分數: 0 - 10
應用App Inventor編程環境中的元件和指令方塊，以及程式的預期功能（20%）	在需要解決問題的情境中，能出色地利用App Inventor編程環境中的元件和指令方塊，完全達致預期效果。 分數：16 - 20	在需要解決問題的情境中，能善用 App Inventor 編程環境中的元件和指令方塊，大致達致預期效果。 分數：11 - 15	在需要解決問題的情境中，只有小部分能恰當地運用App Inventor元件和指令方塊，創建出合適的程序，只有小部分達致預期效果。 分數：6 - 10	在需要解決問題的情境中，只能跟隨指示 / 示範，有限地運用App Inventor元件和指令方塊創建程序，當中只有個別程序能達致預期效果。 分數: 0 - 5

專題習作評分標準 (小組)

美觀及創意，以及使用者的互動體驗（12%）	使用者界面充分照顧使用者的需要，出色地利用「水平配置」和「垂直配置」，以及多個元件，並有邏輯和美觀地安排使用者界面，創建出互動豐富和能讓使用者積極參與的程序，能令使用者從開始到結尾都投入其中，帶來非常優質的體驗。 分數：10 - 12	使用者界面大致能照顧使用者的需要，有運用多個元件，並有邏輯和美觀地安排使用者界面，創建出能讓使用者帶來具參與性的程序，多能令使用者投入其中，只有個別 / 小部分操作或介面設計可以再改善。 分數：5 – 9	使用者界面較為基本，運用簡單元件，嘗試合理地安排使用者界面，當中使用者的互動體驗一般，不少操作或介面設計不清晰，需要改善。 分數：4 - 6	使用者界面不容易使用、指示不清晰、或用法跟日常生活情境的步驟不相似，未能令使用者投入或順暢地使用。當中使用者的互動體驗不佳，需要改善。 分數： 0 - 3
與同儕合作（12%）	團隊成員之間的溝通和協作非常有效，有強烈的團隊精神。團隊成員能通力合作，坦誠溝通，有效令各人發揮所長並完成任務。每位團隊成員都有明確的角色，表現出高度的信任和尊重。 分數：10 - 12	團隊成員之間的溝通和協作有效，有良好的團隊精神。團隊成員相互合作並完成任務。每位團隊成員都有不同的角色，表現出信任和尊重。 分數: 5 - 9	大部分團隊成員能溝通和協作，但個別團隊成員獨立工作。大部分團隊成員有不同的角色，嘗試透過彼此合作，完成任務。 分數: 4 - 6	團隊成員的溝通和協作有限，缺乏團隊精神。團隊成員多為獨立工作，未能互相幫助解決問題，影響協作成效，甚至使任務未能完成。 分數: 0 - 3
總分 100 **優異：76 - 100　　良好：51 - 75　　尚可：26 - 50　　繼續努力：0 - 25**				

專題習作評分標準 (個人)

範疇	優異	良好	尚可	繼續努力
定義問題（20%）	能透過仔細的觀察和具體的分析來清楚地辨識和定義專題習作需解決的問題，從而提出適切可行及有效的解決方案 / 設計，並制定周詳計劃來解決該問題。 分數：16 - 20	大致能透過觀察和分析來辨識和定義專題習作需解決的問題，且提出可行的解決方案 / 設計，並制定合適的計劃來解決問題。 分數：11 - 15	嘗試辨識和定義專題習作需解決的問題，且提出一些可行的解決方案 / 設計，並制定簡單計劃來解決當中的小部分問題。 分數：6 - 10	未能辨識和定義專題習作需解決的問題，提出的解決方案 / 設計只能解決當中的個別問題。 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，命名清晰，能出色地運用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，甚至加入清單（list）或程序（procedure），從而進一步簡化和優化程序，高效及有系統地完成所有任務。 分數：31 - 40	程序設計中大致能呈現合乎邏輯的序列（sequence）。程序以有效的方式設計，命名清晰，多能善用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，從而簡化和優化程序，完成大部分任務。 分數：21 - 30	程序設計中部分能呈現合乎邏輯的序列（sequence），大致能恰當運用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，完成小部分任務。 分數：11 - 20	程序設計中較少能呈現合乎邏輯的序列，程序中少能正確運用循環（iteration）、分支 / 選擇（branching / selection）和變量（variables）等，任務大多未能完成。 分數: 0 - 10
應用App Inventor編程環境中的元件和指令方塊，以及程式的預期功能（20%）	在需要解決問題的情境中，能出色地利用App Inventor編程環境中的元件和指令方塊，創建出互動和讓使用者積極參與的程序，完全達致預期效果。 分數：16 - 20	在需要解決問題的情境中，能善用App Inventor 編程環境中的元件和指令方塊，大致達致預期效果。 分數：11 - 15	在需要解決問題的情境中，只有小部分能恰當地運用App Inventor元件和指令方塊，創建出合適的程序，只有小部分達致預期效果。 分數：6 - 10	在需要解決問題的情境中，只能跟隨指示 / 示範，有限地運用App Inventor元件和指令方塊創建程序，當中只有個別程序能達致預期效果。 分數: 0 - 5

專題習作評分標準 (個人)

美觀及創意，以及使用者的互動體驗（20%）	使用者界面充分照顧使用者的需要，出色地利用「水平配置」和「垂直配置」，以及多個元件，並有邏輯和美觀地安排使用者界面，創建出互動豐富和能讓使用者積極參與的程序，能令使用者從開始到結尾都投入其中，帶來非常優質的體驗。 分數：16 - 20	使用者界面大致能照顧使用者的需要，有運用多個元件，並有邏輯和美觀地安排使用者界面，創建出能讓使用者帶來具參與性的程序，多能令使用者投入其中，只有個別 / 小部分操作或介面設計可以再改善。 分數：11 - 15	使用者界面較為基本，運用簡單元件，嘗試合理地安排使用者界面，當中使用者的互動體驗一般，不少操作或介面設計不清晰，需要改善。 分數：6 - 10	使用者界面不容易使用、指示不清晰、或用法跟日常生活情境的步驟不相似，未能令使用者投入或順暢地使用。當中使用者的互動體驗不佳，需要改善。 分數： 0 - 5
總分 100 **優異：76 - 100　　良好：51 - 75　　尚可：26 - 50　　繼續努力：0 - 25**				

Seven-Step Guide of TPACK in CTE

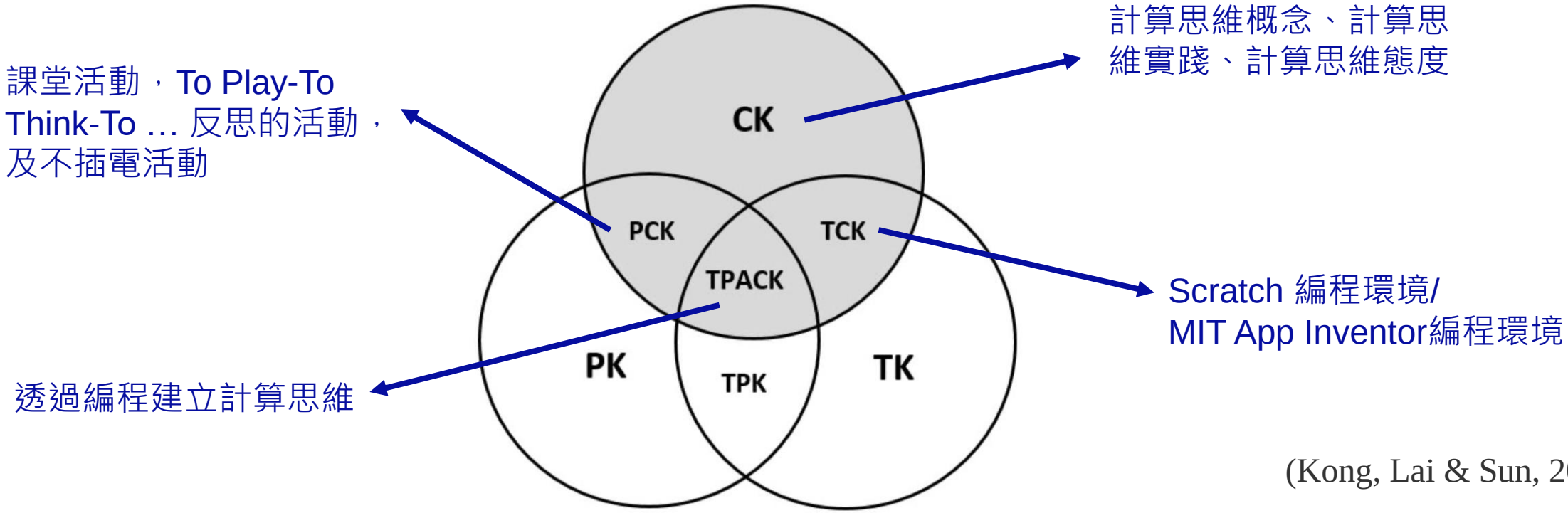
計算思維教育 (CTE) 的科技教學學科知識(TPACK)七步曲

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

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。



(Kong, Lai & Sun, 2020)

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)

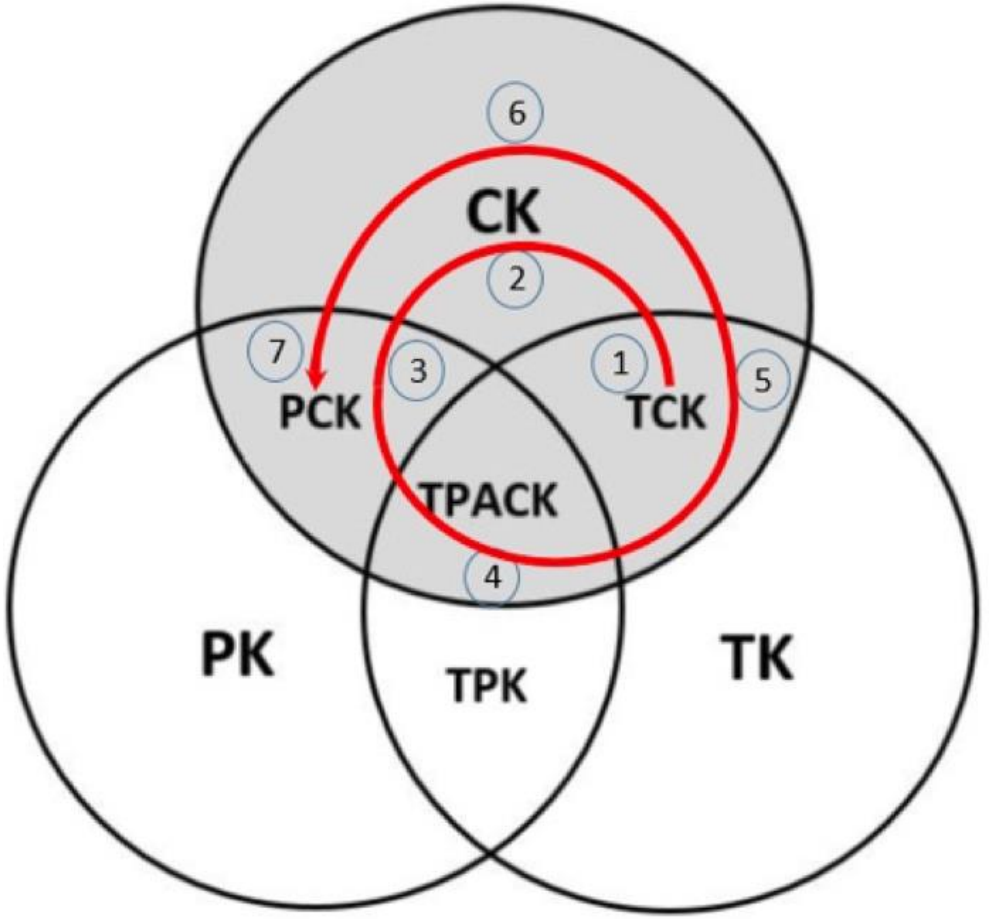
第三步: PCK (教學內容知識)

第四步: TPACK (科技教學學科知識)

第五步: TCK (數碼創意)

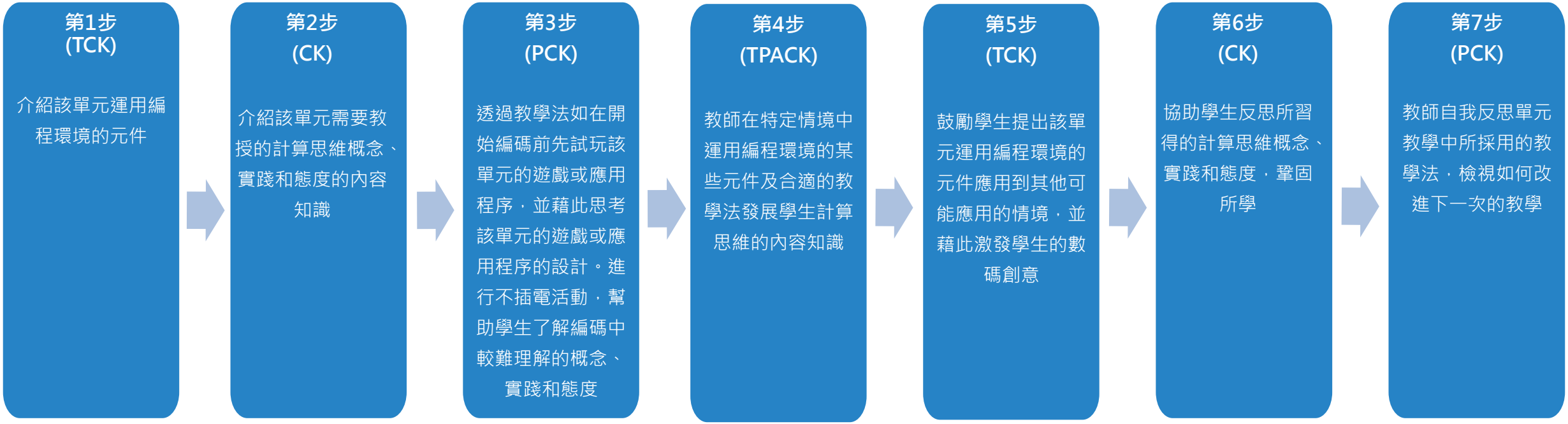
第六步: CK (總結計算思維概念、實踐和態度)

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



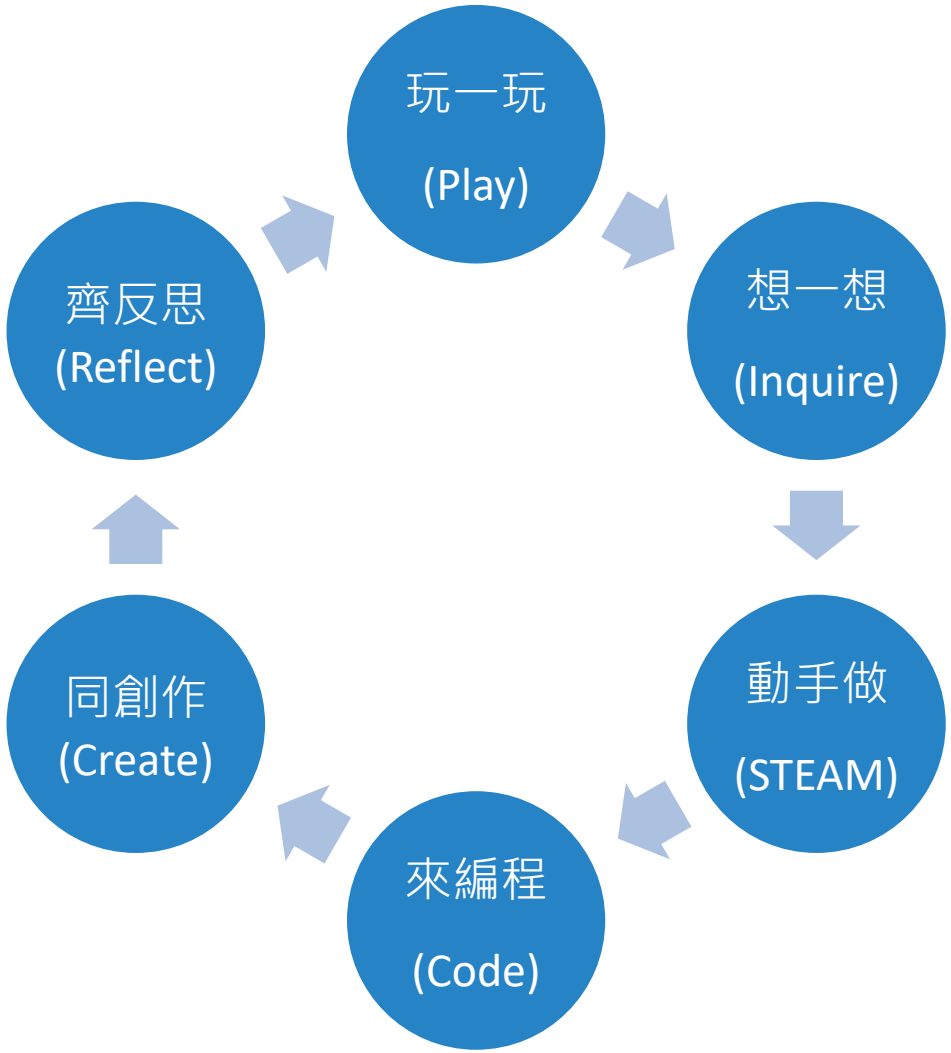
Teachers can flexibly arrange the steps to groom students' problem-solving skills and digital creativity.

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。

A six-step STEAM pedagogy

STEAM 教學法六步曲

STEAM 教學法六步曲



(Kong, 2024)

STEAM 教學法六步曲

第1步
玩一玩

供學生試玩單元的遊戲，讓他們初步理解系統的運作，為他們確立學習目標。

第2步
想一想

學生思考遊戲過程，拆解並探索系統的各個部分。

第3步
動手做

學生動手組裝不同部件來構建系統。通過解決問題，學生能培養因果推理及工程系統思維等解難能力。

第4步
來編程

學生進行編程，從而學習「序列」和「分支 / 選擇」等重要編程概念。

第5步
同創作

鼓勵學生提出該單元運用STEAM的概念應用到其他情境，並藉此激發學生的數碼創意。

第6步
齊反思

協助學生對同儕及自我進行反思，為對方的作品提供具建設性的回饋。

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。

教師自我反思

如何在教學和專題研習中，培育學生的解難能力、數碼創意及堅毅精神、正向價值觀？

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

References

教育局。(2020)。*計算思維－編程教育：小學課程補充文件 (修訂版)* 。中華人民共和國香港特別行政區教育局。

Education Bureau. (2020). *Computational Thinking – Coding Education: Supplement to the Primary Curriculum*. Education Bureau, Hong Kong Special Administrative Region of the People's Republic of China.

Kong, S.C. (2024). Pedagogical design of STEM activities for developing problem-solving skills and digital creativity of primary students in the internet of things era: Six-step STEM pedagogy. In W.M. So, & Z.H. Wan, & T. Luo (Eds.), *Cross-disciplinary STEM learning for Asian primary students: Design, practices and outcomes.* NY: Routledge.

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

References

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

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Huynh, Q. T., Nguyen, U. D., Irazabal, L. B., Ghassemian, N., & Tran, B. Q. (2015, April 9). Optimization of an accelerometer and gyroscope-based Fall Detection Algorithm. Journal of Sensors. https://www.hindawi.com/journals/js/2015/452078/

STEAM教育學與教和評估系列：「高小增潤編程教育課程單元－小六」簡介會(新辦) 回饋問卷

https://forms.gle/6QLWSPKq7yw6veDB8

1. 此簡介會加深了我對計算思維的認識。
2. 此簡介會加深了我對編程教育的教學法和設計學與教材料的了解，促進學生計算思維能力的發展。
3. 總的來說， 我對這活動感到滿意。
4. 活動的目標可以達到。
5. 活動內容切合主題。
6. 總的來說， 導師的整體表現有效。
7. 其他意見（如有）：_____

Briefing Session on the "Enriched Module on Coding Education for Upper Primary Level – Primary 6" (New)

STEAM教育學與教和評估系列：「高小增潤編程教育課程單元 – 小六」簡介會(新辦)

香港教育大學 江紹祥教授
Professor Kong Siu-Cheung,
The Education University of Hong Kong

11 July 2024 二零二四年七月十一日