

Swift Playgrounds Teacher Training

Trainng Topics

Basic Logics

Example :

如果 考第一名 就會 有獎學金

的反過來的邏輯

沒有 獎學金 就 不是第一名

Swift lang relpacement :

```
if 考第一名 {  
    有獎學金  
}
```

```
if !獎學金 {  
    !第一名  
}
```

Addons

Example 1:

如果 考第一名 而且 有準時吃飯 就會 有獎學金

Swift lang replacement :

```
if 考第一名 && 有準時吃飯 {  
    有獎學金  
}
```

的反過來的邏輯是？

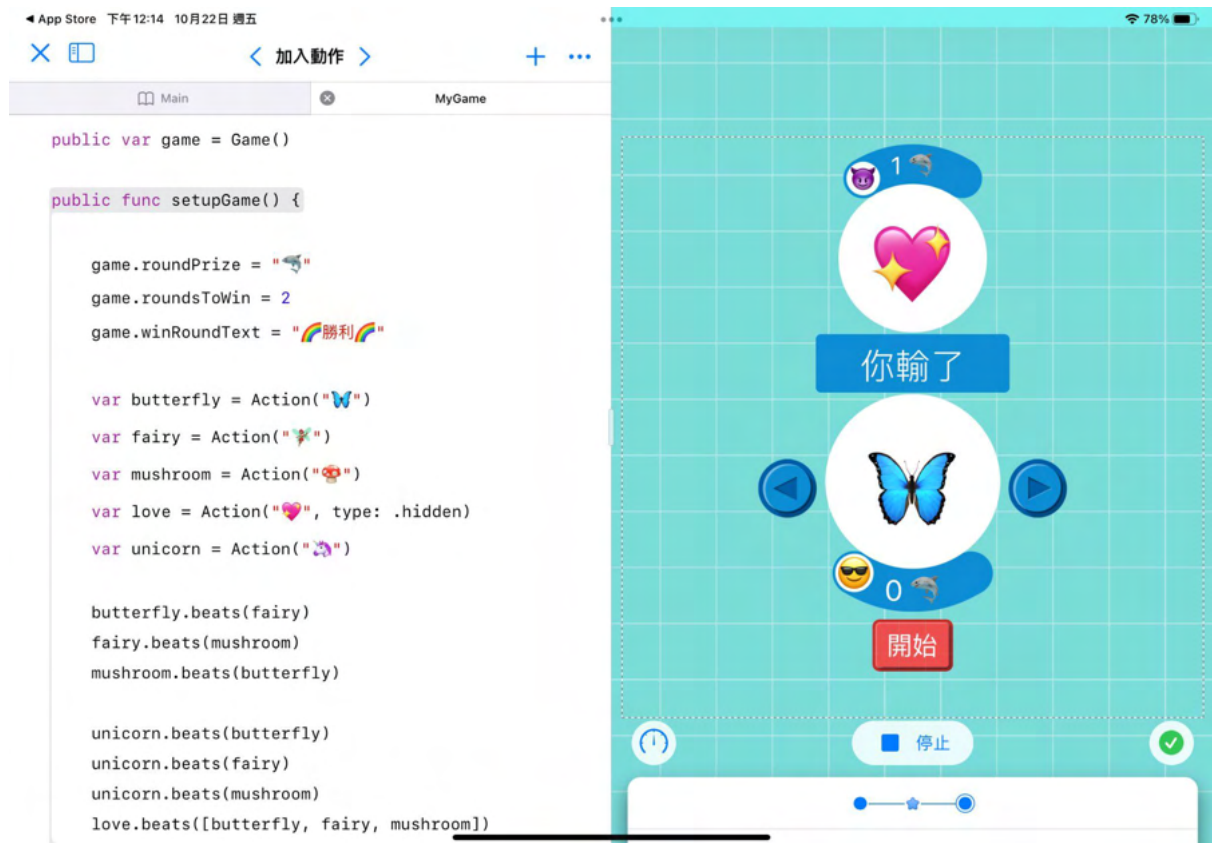
Exampel 2:

如何建立 Function 可以處理 collectGem() 和 toggleSwitch() 如下圖情況？



Imitation

如何建立一個自己的類似剪刀石頭布的遊戲



Problem Solving

Think and create your own algorithm.

◀ App Store 下午 12:16 10月22日 週五

× ⓘ < 簡介 > + ...

Main SharedCode

一種讓人物從地圖上的 A 點走到 B 點的 App，使用尋找路線演算法可以找出穿越小鎮的最快路線，也可以在遊戲中讓壞人追逐英雄。在這個 Playground 中，你將探索尋找路線演算法，用它來破解由地磚和牆磚組成的迷宮。

一些尋找路線演算法相比起來效率更高，因為它們搜尋較少的迷宮區域來找出路線。另一些則稍慢，但能找出最短路線。這個 Playground 探索了兩種設計：廣度優先搜尋和 A*（讀作「A 星」）。

在你的程式碼中，你可以查看一塊磚是地磚、牆磚還是目標磚。還可以搜尋某塊磚相鄰的四塊磚（上下左右的磚塊），直到找到目標磚。

下方的程式碼雖然不能破解所有迷宮，但為你展示了各個類型、方法和屬性。在接下來的頁面中，你需要利用這些資訊來編寫破解迷宮的程式碼。

在即時顯示區中，透過呼叫 `searchNeighbors(of:)` 搜尋的磚塊會變成綠色。程式碼執行後，你會看到已搜尋的磚塊數以及找出的路線長度。

```
func findPath(in maze: Maze) {  
    // 建立座標佇列並加入迷宮的起點座標。  
    var queue = CoordinateQueue()  
    queue.add(maze.start)  
  
    // 持續搜尋座標，直到清空佇列。  
    while !queue.isEmpty {  
        let coord = queue.popFirstCoordinate()  
  
        // 取得第一個鄰近座標。  
        let neighbors = maze.searchNeighborsFor:
```

搜尋次數 5 路線長度 6

▶ 執行我的程式碼

Creating

下午 12:18 10月22日 週五



Flash Card - Final



78%

```
import SwiftUI
import PlaygroundSupport

struct Content : View {
    let questions =
        ["這是猴子嗎?", "這不是貓嗎?", "這可能是大象?"]
    let answers = [true, false, true]
    let images = [Image(uiImage:
        🐵), Image(uiImage: 🐱), Image(uiImage: 🐘)]
    @State var questionCount = 0
    @State var correctCount = 0
    var body : some View {
        VStack{
            Text("總共 \(questions.count)
                題").font(.system(size: 30))
            Text("答對 \(correctCount) 題
                ").font(.system(size: 30))
            if questionCount != questions.count {
                Text(questions
                    [questionCount]).font(
                    .system(size: 50))
                images
                    [questionCount].resizable()
                    .scaledToFit()
```



■ 停止

Training Courses

Pre-requirement learning : (Basic Swift commands) 指定老師
參加前要先過關。每個 Project 指定內容不同

Project A : Pre-learning Q&A + Sphero 垃圾收集器 - 3 hr

Pre-learning Topics

Learn to code 1 : 指令, 函式, For 迴圈。

學習重點

指令

認識操作介面

學習拖拉指令方塊

逐步執行程式碼

定義最少過關指令數

函數(Function)

從一堆可以過關指令中找出最大可重複指令組

將重複出現一連串指令組合出新的 Function

學習命名 Function Name

For

將可以計算次數的指令組用 For 來簡化過關指令數

思考和 Function 的差異

Training 流程

Introduction : 0.5 hr

- 介紹教學用電子書

- 如何設計課程

- Pre learning QA Time

Robot basic : 1 hr

- 利用 訂閱 URL 下載 Sphero 用書 Arcade 1

(URL 由講師提供)

- Arcade 1 練習

- Roll

如何讓 Sphero 滾動

- Aim

學習什麼是程式控制硬體的移動和停止和軟體遊戲的移動停止的差異
當硬體接觸到指令時，需要有停止接收指令來區分不同移動指令

- Heading

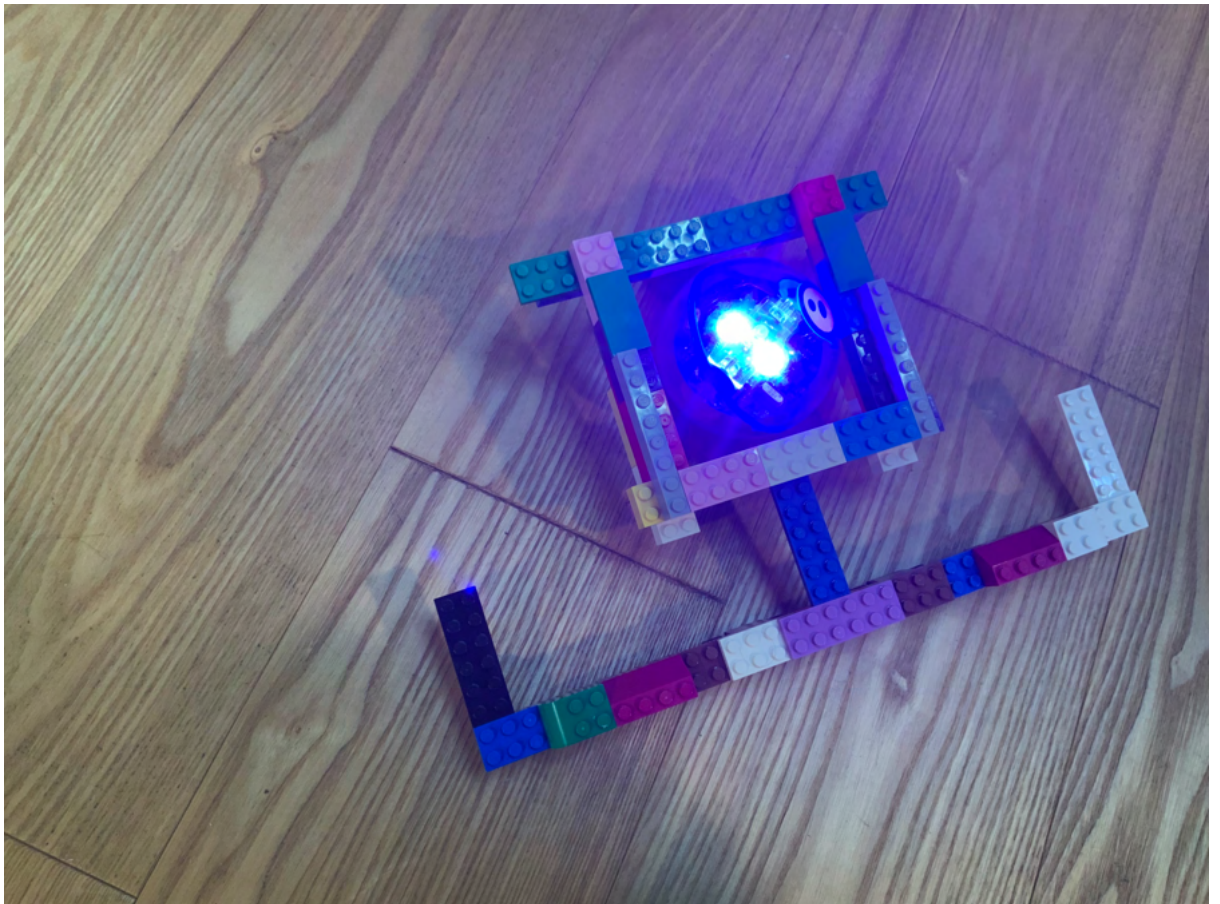
了解什麼是 Sphero 的頭和尾

- Combine For & Function

利用以上所學畫出幾何圖形

Project setting : 0.5 hr

-- 手動利用身邊可用物件 (吸管 or 樂高) 組合出, 可以由 Sphero 拖拉收集垃圾之延伸如下圖所示



Mission : 0.5 hr

- 在地面上灑上幾個乒乓球，讓學員分組可以寫出，最有效率收集所有乒乓球的程式

- 除了改寫程式之外，也可以改變組裝方式

Conclusion : 0.5 hr

-- 反思, 回饋

Project B : Pre-learning Q&A + 防疫送餐機器人 - 3 hr

Pre-learning Topics

Learn to code 1 : 條件碼, 邏輯運算子, While 迴圈, 演算法

學習重點

條件碼

if else-if else 和現實生活中邏輯判斷的對應

如何用勾選快速產生 else-if 和 else

邏輯運算子

當多個不同情況發生時如何利用 AND (&&) OR (||) NOT (!) 組合出正確的邏輯

While 迴圈

可以直接理解成重複很多次的 if

這些關卡主要是練習觀察可以符合過關條件的 while loop

有些關卡只有一種寫法可以參考解答, 不用探究一開始想不出來

演算法

結合 Learn to code 1 所有語法的綜合練習

if else 的練習最為重要

其次為 function 的產生

For - loop

While loop 最不好想

QA : 0.5 hr

Robot basic : 1 hr

- 利用 訂閱 URL 下載 Sphero 用書 Arcade 1 & 2

- Arcade 1 練習

- collisions

熟悉事件觸法的程式碼

- Arcade 2 練習

- Tap

什麼是 Sphero 的 Tap 手勢操作

- Toss

什麼是 Sphero 的 Toss 手勢操作

- Spin

什麼是 Sphero 的 Spin 手勢操作

- Shake

什麼是 Sphero 的 Shake 手勢操作

Project setting : 0.5 hr

- 利用吸管, 紙板, 樂高, 可以做出可以承載東西的托盤

Mission : 1 hr

- 情境發想: 利用 Sphero 拖拉托盤自動送物品給某個隔離房間的人
- 學員需要規劃好房間路線並準備好 Sphero 和托盤

Mission 1 當 Sphero 啟動時, User 利用 Sphero 手勢 (Tap, Toss, Spin or Shake) 會自動把托盤拉到規劃好的房間

Mission 2 當房間的人收到物品, 再利用某個 Sphero 手勢 (Tap, Toss, Spin or Shake) 會自動回到原點。

Project C: 感測器運用, 聲音彈跳器, 製作自己的音準練習 or 調聲器 - 3 hr

Pre-learning Topics

Learn to code 2 : 變數, 類型, 陣列

Blu's adventure : 座標

QA : 1 hr

Program basic : 1 hr

Mission: 1 hr

Project D: 感測器創作, 拍拍魚。用拍手 or 鼓棒打擊來完成一個節奏遊戲 共 3.5 小時

Pre-learning Topics

Learn to code 2 : 初始化, 參數。

Blu's adventure : 觸控事件, 事件處理器。

QA : 1 hr

Program basic : 1.5 hr

Mission : 1 hr

Project E: SwiftUI 利用 Flash Memory Card 的機制建立一個的 App - 4 hr

Pre-learning Topics

光線, 相機, 程式碼 : 全部

組裝你的相機 : 全部

(Optional) Web Resource :

<https://developer.apple.com/tutorials/swiftui/creating-and-combining-views>

QA : 1 hr

Program basic : 2 hr

Complete rest codes : 1 hr

Project F: SwiftUI 建立一個自己的 Youtube App - 4 hr

Pre-learning Topics

華麗的照片 : 全部

Web Resource :

<https://betterprogramming.pub/write-swiftui-entirely-on-an-ipad-using-swift-playgrounds-f48364e50c9a>

(Optional) <https://www.hackingwithswift.com/quick-start/swiftui>

QA : 1 hr

Program basic : 2 hr

Completer rest codes : 1 hr