

新北市115年度國中小資訊科技優良教案徵選實施計畫

教案設計

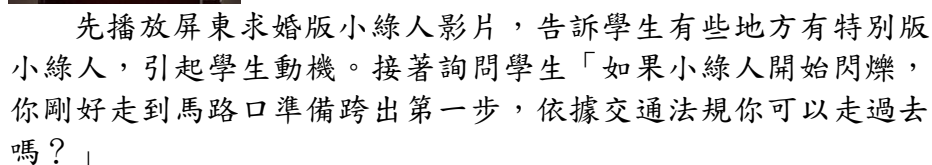
服務學校	崇林國中		設計者	沈鈺淋
參加組別	☒ 程式教育組 ☐ 人工智慧組 ☐ 資訊素養與倫理組			
領域/科目	資訊科技		實施年級	七年級
單元名稱	基礎程式設計		總節數	共__4__節，_180_分鐘
設計依據				
學習重點	學習表現	能應用運算思維解析問題。 能選用適當的資訊科技組織思維，並進行有效的表達。 能設計資訊作品以解決生活問題。	核心素養	系統思考與解決問題 藝術涵養與美感素養 規劃執行與創新應變
	學習內容	程式語言基本概念、功能及應用 陣列程式設計實作 模組化程式設計與問題解決實作 資訊科技應用專題		
議題融入	實質內涵	了解日常生活容易發生事故的原因。		
	所融入之學習重點	認識標誌標線與號誌，理解智慧交通與大眾運輸系統。		
與其他領域/科目的連結	無			
教材來源	翰林課本、Michael Pan Swift App 入門實作			
教學設備/資源	平板、鏡像投影			
使用軟體、數位資源或 APP 內容	Swift playground			
學習目標				
1. 學生能從編寫應用程式中，理解三大結構。 2. 學生可以在設計介面與按鈕功能中，運用流程圖解決問題。 3. 學生能運用陣列解決問題。 4. 學生能拆解應用程式中按鈕按下去之後的動作。 5. 小組討論時，學生可以相互合作並完成小組任務及上台展示及報告。				

教學活動內容及實施方式

使用軟體、數位
資源或 APP 內容

課前活動(學生自學):學生應先預習，課本 P.170-173，關於流程圖部分。

一、引起動機



引導思考與觀念澄清：學生常誤解以為閃爍只是提醒走快一點，所以衝過去沒關係。「綠燈閃爍」代表時間已經不足以讓未進入馬路的行人安全通過。如果還沒踏上斑馬線，不可以強行通過。

今天寫的程式，就要實作紅綠燈燈號排列。

環境架構與基本排版介紹

Youtube

Swift Playground
無邊際

```

1 import SwiftUI          // 匯入 SwiftUI，有畫面和按鈕
2 import AVFoundation     // 匯入語音功能，讓手機可以說話
3
4 // 這是我們的主要畫面
5 struct ContentView: View {
6
7     @State var isGreen = false    // 號誌是不是綠燈
8     @State var seconds = 0        // 綠燈秒數
9
10    // 控制小綠人的動畫
11    @State var stepToggle = false // 小綠人圖片切換
12    @State var blink = false      // 三秒閃爍切換
13
14    let speaker = AVSpeechSynthesizer() // 建立語音
15
16    // 主畫面內容
17    var body: some View {
18        VStack() { // 垂直排列
19
20            // 畫紅綠燈
21            Circle()

```

import 匯入模組

VStack { } 會從上到下垂直堆疊排好，這就是循序結構

@State 它叫做狀態變數，只要變數改變，畫面會跟著改變



接著請學生完成 任務一：循序結構區塊

```

isGreen = true          // 第一步：切換成綠燈
seconds = 10            // 第二步：倒數設定為 10 秒
stepToggle = false      // 第三步：小綠人出第一隻腳
blink = false           // 第四步：閃爍開關預設關閉

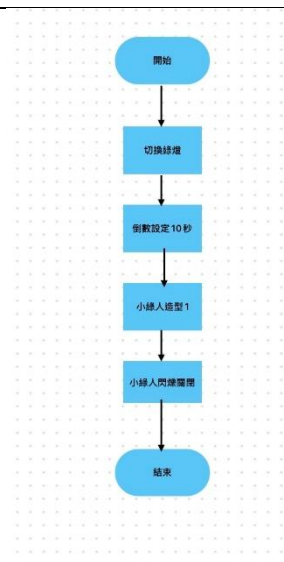
```

三、組內共學

1. 流程圖練習，小組討論並在平板的無邊際上畫出這個初始化設定的「紅燈-小綠人」的循序流程圖。

2. 完成按鈕，按下按鈕，紅燈變綠燈且出現「剩餘 10 秒」文字，表示 4 行程式碼依序執行了。

20 分鐘



ContentView

```

43
44 // 開啟綠燈函式
45 func startGreenLight() {
46 // -----
47 // 【第一節課任務：填入循序結構區塊】
48 // -----
49
50
  
```

四、組間互學

教師指示：左右兩兩組別交換學習單，檢查兩點：

1. 有沒有「開始」與「結束」？
2. 箭頭方向有沒有畫錯？
3. 順序有沒有錯誤？

2-3-1 循序結構

程式若是由上而下依序執行，就稱為循序結構。

循序結構－流程圖



若有組別需訂正，老師可投影畫面與全班討論

下節課預告：請學生回家預習 P. 196，關於選擇結構

5 分
鐘

2-3-2 選擇結構

依條件判斷的結果，決定要執行哪一段程式碼敘述。條件判斷的結果，有條件成立（true）及條件不成立（false）兩種。

在單向選擇結構中，只定義了條件成立時要執行的敘述；在雙向選擇結構中，分別定義了條件成立與條件不成立時要執行的敘述。

單向 選擇結構—流程圖

例如：使用單向選擇結構的 Scratch 程式碼

判斷條件：條件是八角形積木。

如果條件成立，就執行輸入程式。

雙向 選擇結構—流程圖

例如：使用雙向選擇結構的 Scratch 程式碼

判斷條件：條件是八角形積木。

如果條件成立，就執行輸入的上方程式；如果條件不成立，就執行輸入的下方程式。

第二節課

一、引起動機

「上一節課我們做好了靜態的排版。現在我們要讓按鈕派上用場！有些紅綠燈設有行人通行按鈕，按下『切換號誌鈕』後，程式怎麼知道該開綠燈還是保持紅燈？這就需要課本的：選擇結構！」

點擊按鈕，畫面為什麼沒有變綠燈？「因為按鈕函式裡面是空的，我們要完成這裡讓他可以切換。」

完成按鈕與切換綠燈函式 任務二：判斷結構區塊



剩餘 9 秒

切換號誌

```
if seconds > 0 {
    // 條件成立（倒數中）：什麼都不做，鎖定按鈕！

} else {
    // 條件不成立（秒數等於 0）：才執行開綠燈
    startGreenLight()
}
```

10分
鐘

課本

```

31
32         // 4. 切換號誌按鈕
33         Button("切換號誌") {
34             // -----
35             // 【第二節課任務：填入判斷結構區塊】
36             // -----
37             startGreenLight()
38         }
39         .buttonStyle(.bordered)
40     }
41     .padding()
42 }
43

```

二、組內共學

按鈕裡面的程式碼：

```
if seconds > 0 { } else { startGreenLight() }
```

請學生討論並思考：為什麼當剩餘秒數大於 0 時，我們不讓按鈕重新啟動綠燈？」

因為如果小綠人走到一半，路人一直按按鈕讓時間重新歸零

10分
鐘

Swift
Playground
學習單

三、教師導學

老師補充說明，當時間來不及過時要閃爍。觀察這行控制小綠人透明度的程式碼：

```
.opacity(!isGreen ? 0.2 : (seconds <= 3 && blink ? 0.2 : 1))
```

這裡的 &&，是「而且 (And)」，小綠人要變淡(閃爍)，必須同時滿足兩個條件：「秒數剩下 3 秒以下」而且「閃爍開關 blink 目前是 true(打開)」。兩個條件同時成立，透明度才會變成 0.2。

10分
鐘

Swift
Playground

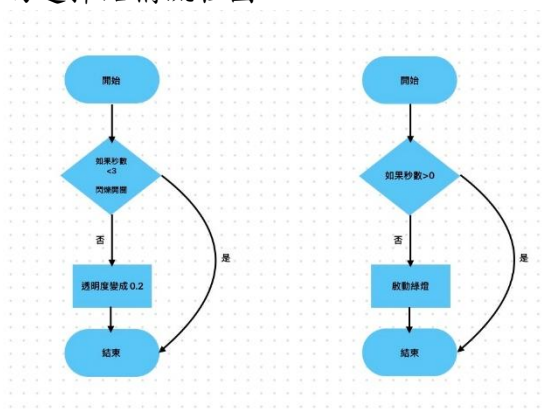
四、組內共學

1. 流程圖練習，小組討論並在平板的無邊際上畫出這個「切換號誌」的選擇結構流程圖。

2. 流程圖練習，小組討論並在平板的無邊際上畫出這個「燈號閃爍」的選擇結構流程圖

15分
鐘

Swift
Playground
無邊際
學習單



下節課預告:請學生回家預習 P. 200，關於重複結構

2-3-3 重複結構

需要重複執行的程式碼可以使用重複結構，根據被重複執行的情況，可以分為計次式迴圈與條件式迴圈兩種。

當我們明確知道某段敘述要被重複執行的次數時，就會使用計次式迴圈；當我們不知道某段敘述要被重複執行的次數，而是知道終止情況時，就會使用條件式迴圈。

小知識
迴圈
在一段程式語言內，可以讓程式碼反覆執行的敘述。

計次式迴圈一流程圖

例如：使用重複結構的 Scratch 程式碼

第三節課

一、引起動機

「小綠人走動需要一直更換圖片。如果綠燈 10 秒，難道我們要手動寫 20 次更換圖片的程式碼嗎？這時候就要靠重複結構！」

解說課本重複結構如何運作。

二、教師導學

完成任務三：迴圈結構區塊

```

Timer.scheduledTimer(withTimeInterval: 0.5, repeats:
true) { timer in
    stepToggle.toggle() // 每 0.5 秒切換換腳圖案

    // 最後 3 秒觸發閃爍
    if seconds <= 3 && isGreen {
        blink.toggle()
    }

    // 每換腳兩次（1 秒）扣一次秒數
    if stepToggle {
        seconds -= 1
    }

    // 跳出迴圈的終止條件
    if seconds <= 0 {
        timer.invalidate() // 關閉迴圈，避免當機
        isGreen = false
        blink = false
    }
}

```

5 分
鐘

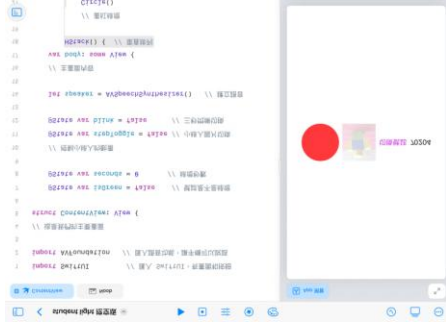
課本

10分
鐘

Swift
Playground
無邊際

<pre> } } </pre> 三、組內共學 <pre> 49 50 51 // ----- 52 // 【第三節課任務：填入迴圈結構區塊】 53 // ----- 54 55 } </pre> 1. 完成迴圈並測試結果：按下按鈕，小綠人開步行走，最後三秒自動閃爍，歸零後自動變紅燈停止！ 2. 流程圖練習，小組討論並在平板的無邊際上畫出這個「倒數直到秒數結束」的重複結構流程圖 <pre> graph TD Start([開始]) --> LoopStart{重複直到秒數=0} LoopStart -- 否 --> Switch[切換圖片] Switch --> Dec[秒數扣1] Dec --> Flash{如果秒數<3 持續閃爍幾秒} Flash --> Trans[透明度變成 0.2] Trans --> SetFalse[綠燈=false 閃爍=false] SetFalse --> End([結束]) LoopStart -- 是 --> LoopStart </pre> 四、組間互學 時間有限，可以只請幾組學生上台解說自己組別畫出來的流程圖額外加分。 五、教師總結 強調 <code>timer.invalidate()</code> 的重要性：如果沒有這行，計時器會永遠跑下去，造成無法停止的無窮迴圈。 第四節課	20分 鐘 5分 鐘 5分 鐘	Swift Playground 無邊際 學習單 Swift Playground 無邊際 Swift Playground Youtube
---	---	--

一、引起動機 「我們已經完成了一個會走動、會閃爍、會倒數的小綠人。但如果今天是一位視障人士（盲人）要過馬路，他看不到怎麼辦？科技必須無障礙空間設計。台灣有些馬路旁有『有聲號誌』，我們今天要讓我們的小綠人有聲音！」 給學生看視障友善的馬路影片： https://youtu.be/JQ2WZ4t0DKg?si=wjCntNvA5zZFam-X	5 分鐘	
二、教師導學 import AVFoundation：匯入影音工具箱。 AVSpeechSynthesizer()：建立語音發聲器。 AVSpeechSynthesisVoice(language: "zh-TW")：設定講台灣中文示範 func speak(_ text: String)，只要輸入不同的文字字串，手機就會說出來。 <pre> 55 } 56 57 // ----- 58 // 【第四節課任務：填入 speak 語音函式區塊】 59 // ----- 60 61 } 62 </pre> 完成任務四：語音函式區塊 <pre> func speak(_ text: String) { let utterance = AVSpeechUtterance(string: text) utterance.voice = AVSpeechSynthesisVoice(language: "zh-TW") speaker.speak(utterance) } </pre>	5 分鐘	Swift Playground
三、組內共學 1. 修改語音文字，加入各組創意（例如：「綠燈亮起，台南牛肉湯提醒您小心通行」） 2. 測試完整程式：點擊按鈕 → 語音說「綠燈亮起，請小心通行」 → 小綠人走路 → 最後三秒閃爍 → 倒數結束變紅燈 → 語音說「紅燈亮起，請停止」	15分鐘	Swift Playground 無邊際學習單
四、組間互學 每組 2 分鐘，展示完整程式及創意語音，各組填寫同儕互評表（滿分 5 星）。 將程式上傳至學習吧作業區。	15分鐘	Swift Playground 無邊際學習單 學習吧

五、教師總結 循序結構： VStack 內由上而下繪製介面。 選擇結構： if-else 配合菱形流程圖，根據秒數決定是否切換燈號與最後三秒閃爍。 重複結構： Timer 讓小綠人重複換腳，並利用 timer.invalidate() 跳出迴圈避免無窮無盡。	5 分鐘	Swift Playground 無邊際
教學成果		
	說明:學生作品	說明:學生作品
		
教學心得與省思	說明:循序結構解說	說明:學生小組討論
	讓學生分組集思廣益，較有辦法完成完整程式碼。建議即便是分組，也讓每個學生一人一台平板完成程式，才不會有學生無所事事，用給程式鷹架讓學生填充方式，有助於學生學習循序、重複、選擇結構，及流程圖，並能有效降低寫程式一直出現 BUG 的挫折感，課程進行也會比較順利。	
參考資料	https://youtu.be/_eIhtNcWNuc?si=GmvjL7LUoZlzaQEX 屏東小綠人報導 https://youtu.be/JQ2WZ4t0DKg?si=wjCntNvA5zZFAm-X 視障如何過馬路 Michael Pan Swift App 入門實作	

學習單(4頁)

三大結構：小綠人

班級：_____ 組別：_____ 姓名：_____ 座號：_____

任務一：循序結構

在介面設計中，`VStack{}` 會將元件從上到下垂直堆疊排好，這就是程式中的「循序結構」。請根據課堂設定，完成以下小綠人的初始狀態設定。

1. 填空题：請寫出以下變數代表的初始意義：

|

- `isGreen = true` → 第一步：切換成 _____。
- `seconds = 10` → 第二步：倒數設定為 _____ 秒。
- `stepToggle = false` → 第三步：小綠人出 _____。
- `blink = false` → 第四步：閃爍開關預設為 _____。

2. 畫畫看：請與小組討論，並在下方（或平板無邊際 App 中）畫出上述「紅燈-小綠人」初始化設定的循序流程圖。

(提示：記得要有「開始」與「結束」，並注意箭頭方向喔！)

我的流程圖草稿區：

附錄

任務二：選擇結構

當路人按下「切換號誌鈕」時，程式必須判斷現在該不該變成綠燈，這就是「選擇結構」。

1. 思考與討論：

在我們的按鈕程式碼中，設定了以下的條件：

```
if seconds > 0 {  
    // 什麼都不做，鎖定按鈕！  
} else {  
    startGreenLight()  
}
```

請問：為什麼當「剩餘秒數大於 0（倒數中）」時，我們**不讓**按鈕重新啟動綠燈？如果沒有這個限制，行人在過馬路時可能會發生什麼危險或不便？

答：

2. 邏輯解析（&& 的運用）：

控制小綠人閃爍的程式碼中，使用了 &&（而且）。小綠人要變淡閃爍，必須**同時**滿足哪兩個條件？

- 條件一：_____
- 條件二：_____

我的流程圖草稿區：

任務三：重複結構

小綠人走路需要不斷切換圖片，我們利用 `Timer.scheduleTimer` 讓程式每 0.5 秒自動換腳。

1. 除錯大師 (Debug)：

老師在上課時特別強調，當倒數秒數歸零 (`seconds <= 0`) 時，必須執行 `timer.invalidate()`。

請問：這是為了關閉迴圈，如果忘記寫這行程式碼，會發生什麼嚴重的後果？

答：

2. 畫畫看：請與小組討論，畫出「倒數直到秒數結束」的重複結構流程圖。

我的流程圖草稿區：

任務四：無障礙設計與創意發表

科技始於人性！為視障人士設計有聲號誌，我們匯入了 AVFoundation 影音工具箱，並利用 AVSpeechSynthesizer() 建立語音發聲器。

1. 創意發聲：

請發揮小組創意，設計一段專屬的綠燈語音提示（例如：綠燈亮起，台南牛肉湯提醒您小心通行）。

我們設計的語音字串是：

「_____」

2. 同儕互評時間：

請觀摩其他組別的展示（包含：語音創意、小綠人是否正常走路、最後三秒是否閃爍、倒數結束是否變紅燈），並給予評分。

評分項目	觀察重點	給星（滿分 5 星）
程式完整度	小綠人走動、閃爍、紅綠燈切換是否順暢？	★★★★★
創意與巧思	語音提示的設計是否有趣或貼心？	★★★★★
團隊表達	上台展示與報告時是否清楚、有條理？	★★★★★

在空白區給分(請分8等份):

完整程式碼:

ContentView

```

1  import SwiftUI          // 匯入 SwiftUI，有畫面和按鈕
2  import AVFoundation     // 匯入語音功能，讓手機可以說話
3
4  // 這是我們的主要畫面
5  struct ContentView: View {
6
7      @State var isGreen = false    // 號誌是不是綠燈
8      @State var seconds = 0        // 綠燈秒數
9
10     // 控制小綠人的動畫
11     @State var stepToggle = false // 小綠人圖片切換
12     @State var blink = false      // 三秒閃爍切換
13
14     let speaker = AVSpeechSynthesizer() // 建立語音
15
16     // 主畫面內容
17     var body: some View {
18         VStack() { // 垂直排列
19
20             // 畫紅綠燈
21             Circle()

```

ContentView

```

22         .fill(isGreen ? .green : .red) // 綠色or紅色
23         .frame(width: 90, height: 90) // 大小 90x90
24
25         // 小綠人圖片
26         Image(isGreen
27             ? (stepToggle ? "walk1" : "walk2") // 綠燈時交替兩張圖
28             : "walk3")                          // 紅燈時固定一張變淡的圖
29             .resizable()                          // 可以縮放大小
30             .frame(width: 80, height: 90)         // 大小 80x90
31         // 控制透明度
32         .opacity(!isGreen ? 0.2                // 紅燈時變淡
33             : (seconds <= 3 && blink ? 0.2 : 1))
34
35         // 倒數文字，只在綠燈顯示
36         if isGreen {
37             Text("剩餘 \("\(seconds) 秒") // 顯示剩下多少秒
38                 .font(.title)             // 文字大小
39         }
40
41         // 按鈕：切換號誌
42         Button("切換號誌") {

```

ContentView

```
43         if seconds > 0{
44
45             }else{
46                 startGreenLight() // 按下去秒數不大於零，就開始綠燈倒數
47             }
48
49             }.buttonStyle(.bordered)
50         }
51         .padding() // 空格間距
52     }
53
54     // 開始綠燈的函式
55     func startGreenLight() {
56         isGreen = true // 把號誌設成綠燈
57         seconds = 10 // 綠燈倒數 10 秒
58         stepToggle = false // 小綠人切換第一張圖
59         blink = false // 閃爍設定不閃爍
60
61         speak("綠燈亮起，請小心通行") // 說話
62
63         // 設定一個每 0.5 秒跑一次的計時器
64         Timer.scheduledTimer(withTimeInterval: 0.5, repeats: true) { timer in
```

ContentView

```
66         // 腳步交替
67         stepToggle.toggle() // true ⇌ false 換腳
68
69         // 最後三秒閃爍
70         if seconds <= 3 && isGreen {
71             blink.toggle() // 透明度變淡
72         }
73
74         // 每秒扣一次倒數
75         if stepToggle { // 交替一次扣一次秒數
76             seconds -= 1
77         }
78
79         // 倒數結束
80         if seconds <= 0 {
81             timer.invalidate() // 停掉計時器
82             isGreen = false // 紅燈
83             blink = false // 取消閃爍
84             speak("紅燈亮起，請停止") // 語音提示
85         }
86     }
87 }
```

ContentView

```
78
79         // 倒數結束
80         if seconds <= 0 {
81             timer.invalidate() // 停掉計時器
82             isGreen = false    // 紅燈
83             blink = false      // 取消閃爍
84             speak("紅燈亮起，請停止") // 語音提示
85         }
86     }
87 }
88
89 // 語音函式
90 func speak(_ text: String) {
91     let utterance = AVSpeechUtterance(string: text) // 把文字轉成語音
92     utterance.voice = AVSpeechSynthesisVoice(language: "zh-TW") // 用中文
93     speaker.speak(utterance) // 開始說話
94 }
95 }
```

給學生版_挖空程式碼:

ContentView

```
1 import SwiftUI
2 import AVFoundation // (第四節課會用到)
3
4 struct ContentView: View {
5     // 狀態變數
6     @State var isGreen = false    // 燈號狀態
7     @State var seconds = 0        // 倒數秒數
8     @State var stepToggle = false // 小綠人換腳開關
9     @State var blink = false      // 三秒閃爍開關
10
11     // 【第四節課任務】宣告語音合成器 speaker
12
13     var body: some View {
14         VStack { // 循序結構：畫面元件由上而下排列
15             // 1. 紅綠燈圓形
16             Circle()
17                 .fill(isGreen ? .green : .red)
18                 .frame(width: 90, height: 90)
19
20             // 2. 小綠人圖片
21             Image(isGreen ? (stepToggle ? "walk1" : "walk2") : "walk3")
```

ContentView

```
22         .resizable()
23         .frame(width: 80, height: 90)
24         .opacity(!isGreen ? 0.2 : (seconds <= 3 && blink ? 0.2 : 1))
25
26         // 3. 倒數文字
27         if isGreen {
28             Text("剩餘 \ \(seconds) 秒")
29                 .font(.title)
30         }
31
32         // 4. 切換號誌按鈕
33         Button("切換號誌") {
34             // -----
35             // 【第二節課任務：填入判斷結構區塊】
36             // -----
37             startGreenLight()
38         }
39         .buttonStyle(.bordered)
40     }
41     .padding()
42 }
43
```

ContentView

```
43
44         // 開啟綠燈函式
45         func startGreenLight() {
46             // -----
47             // 【第一節課任務：填入循序結構區塊】
48             // -----
49
50
51             // -----
52             // 【第三節課任務：填入迴圈結構區塊】
53             // -----
54
55         }
56
57         // -----
58         // 【第四節課任務：填入 speak 語音函式區塊】
59         // -----
60
61     }
62
```