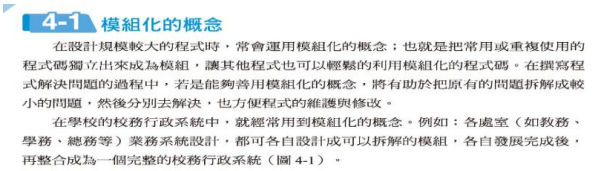
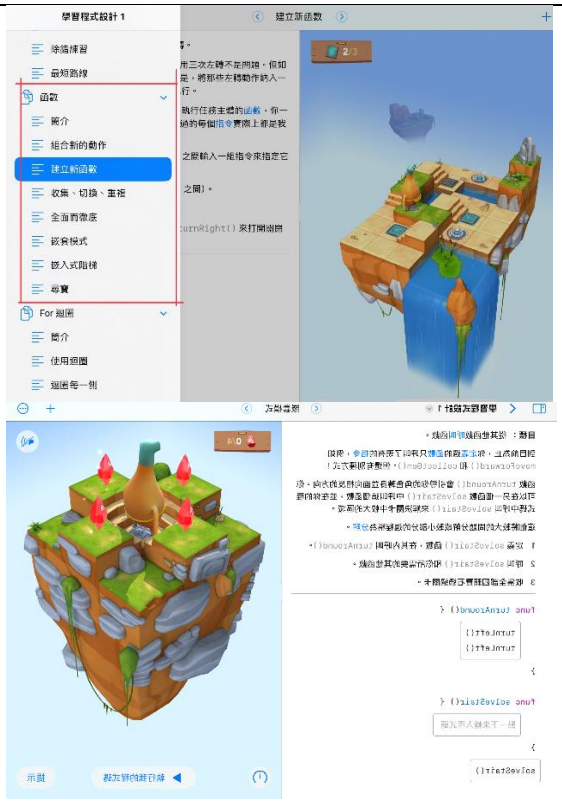
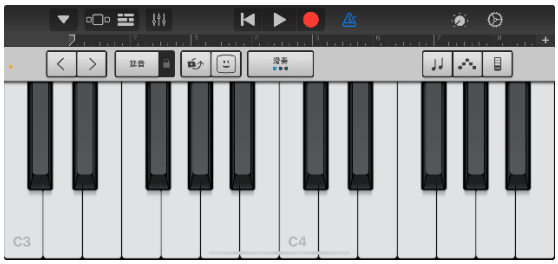
附件2

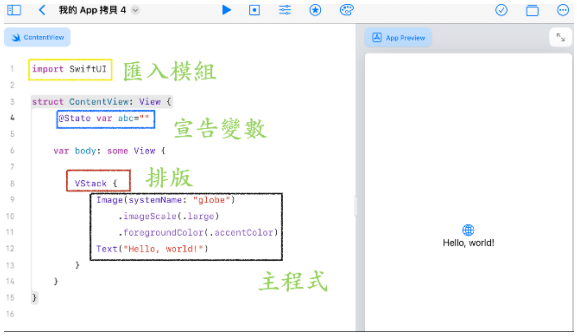
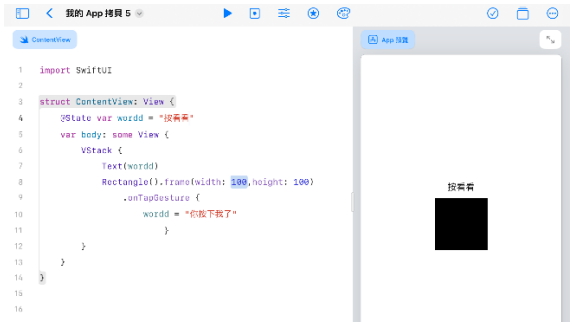
新北市113年度國中小資訊科技優良教案徵選實施計畫

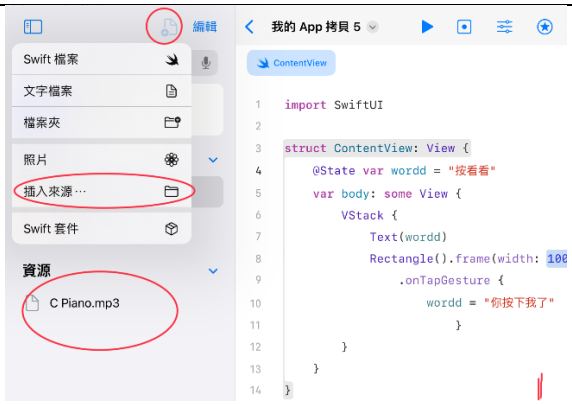
教案設計

<table>
<tr><td>服務學校</td><td colspan="2">崇林國中</td><td>設計者</td><td colspan="2">沈鈺淋</td></tr>
<tr><td>參加組別</td><td colspan="2">■程式教育組　□人工智慧組</td><td colspan="3">□資訊素養與倫理組</td></tr>
<tr><td>領域/科目</td><td colspan="2">科技領域/資訊科</td><td>實施年級</td><td colspan="2">八年級</td></tr>
<tr><td>單元名稱</td><td colspan="2">模組化程式設計</td><td>總節數</td><td colspan="2">共__3__節，_135_分鐘</td></tr>
<tr><td colspan="6">設計依據</td></tr>
<tr><td rowspan="2">學習重點</td><td>學習表現</td><td>運 p-IV-1能選用適當的資訊科技組織思維，並進行有效的表達。
運 t-IV-3 能設計資訊作品以解決生活問題。
設 k-IV-1　能了解日常科技的意涵與設計製作的基本概念。</td><td rowspan="2">核心素養</td><td colspan="2" rowspan="2">科-J-A2系統思考與解決問題
科-J-A3利用科技資源，擬定與執行科技專題活動。
科-J-C2運用科技工具進行溝通協調及團隊合作，以完成科技專題活動。</td></tr>
<tr><td>學習內容</td><td>資 P-IV-3陣列程式設計實作。
資 P-IV-4模組化程式設計的概念。
資 P-IV-5模組化程式設計與問題解決實作。</td></tr>
<tr><td rowspan="2">議題融入</td><td>實質內涵</td><td colspan="4">科 E1 了解平日常見科技產品的用途與運作方式。
科 J1 了解科技本質、科技系統與設計製作的基本概念。
科 J12 運用設計流程，實際設計並製作科技產品以解決問題。</td></tr>
<tr><td>所融入之學習重點</td><td colspan="4">運 p-IV-2 能利用資訊科技與他人進行有效的互動。</td></tr>
<tr><td colspan="2">與其他領域/科目的連結</td><td colspan="4">音樂: 樂器發聲原理與製作</td></tr>
<tr><td colspan="2">教材來源</td><td colspan="4">翰林課本 資訊科技 第四冊 第四章 進階程式設計(2)</td></tr>
<tr><td colspan="2">教學設備/資源</td><td colspan="4">平板、電腦</td></tr>
<tr><td colspan="2">使用軟體、數位資源或APP內容</td><td colspan="4">Swift Playground、Visio、學習吧</td></tr>
<tr><td colspan="6">學習目標</td></tr>
<tr><td colspan="6">1. 學生能了解模組化的應用。
2. 學生能了解副程式的意涵。
3. 學生會使用副程式的參數設定。
4. 學生能繪製出有模組化流程的流程圖。</td></tr>
</table>

教學活動設計		
教學活動內容及實施方式	**時間**	**使用軟體、數位資源或 APP 內容**
一、先備知識學習 (一)先備知識建立與確認: 課前預習:請學生上學習吧，填寫測驗，並觀看測驗結果，錯誤的題目是否可以在上課時找到答案。  第 1 題 單選題 配分 1 模組化的主要目的是？ 1. 提高程式碼行數 2. 減少軟體的可擴展性 3. 降低程式碼重用性 4. 提高軟體的可維護性 正確答案：4 解析 第 2 題 單選題 配分 1 模組化可以把大問題拆成小問題 確保學生對於模組化的認知，所謂模組化就是把大問題拆解成小問題，完成小問題程式模組。透過副程式來達成模組化，主程式能更簡潔、更容易懂大流程。  4-1 模組化的概念 在設計規模較大的程式時，常會運用模組化的概念；也就是把常用或重複使用的程式碼獨立出來成為模組，讓其他程式也可以輕鬆的利用模組化的程式碼。在撰寫程式解決問題的過程中，若是能夠善用模組化的概念，將有助於把原有的問題拆解成較小的問題，然後分別去解決，也方便程式的維護與修改。 在學校的校務行政系統中，就經常用到模組化的概念。例如：各處室（如教務、學務、總務等）業務系統設計，都可各自設計成可以拆解的模組，各自發展完成後，再整合成為一個完整的校務行政系統（圖 4-1）。 老師講述完模組化概念後可以詢問: 1.什麼時候適合使模組化? 大程式裡面經常使用的、重複性高的程式。 2.模組化的優點? 閱讀、除錯較容易，可節省記憶體空間，大程式可以分工合作完成。 (二)模組化練習: 開始進行模組化練習，利用 iPad 的 Swift Playground 程式，用「學習程式設計1」函數章節，讓學生練習模組化。 過程中，學生將會學到拆解問題、定義函數、呼叫函數。	10 **分鐘** **30** **分鐘**	課本、學習吧 平板 Swift Playground

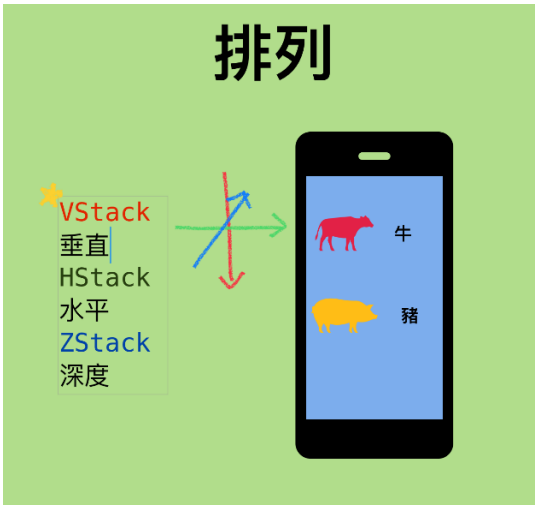
 開始練習時，老師先引導學生拆解問題: 1.沒有右轉的程式該如何右轉? 使用左轉，轉三次即可達成一次右轉。 2.能否利用左轉把右轉作成一個函數? 定義函式:命名:要能從名稱知道函式用途。 3.如何使用函數? 呼叫:在主程式裡面呼叫右轉的副程式。 (三)課後評估: 請學生至學習吧再次填寫測驗，看看使否理解模組化的概念，可以加深印象。	5 分鐘	學習吧
二、模組化的應用(I) (一)情境引導：  開啟 ipad 的 Garageband 找到鋼琴，讓學生體驗在平板上彈鋼琴。 老師先詢問學生問題: 1. 樂器如何發出聲音? 振動及共鳴，根據樂器的弦、管等不同而有不同聲音。	5 分鐘	平板 Garageband

2. 平板上的鋼琴又要如何發出聲音? 點擊之後，根據點擊的按鍵發出相對應的數位化聲音。 3. 透過問題拆解，能否成使用函數完成? 按鍵位置、發出聲音。 (二)分組實作: 3-4人一組，學生分組後，開始繪製流程圖，目標為在平板上製作木琴，討論後將流程與副程式繪製出來，並傳至作業區。 流程圖完成後，老師講授製作木琴會用到的SwiftUI 架構與基本語法。學生主要能學會點擊時觸發以及播放音檔兩個重點語法，排版、修飾等可視時間多寡進行教學。 1. Swift 基本架構與變數設定:  可以先設定一個空值的變數備用: @State var A = “” 2. 建立文字、圖形與按鈕(按鈕觸發): 使用 Text()在畫面上建立文字。 使用 Rectangle()建立一個矩形。使用.frame(height:10,weight:10)設定矩形大小，並在加上.onTapGuesture{}讓矩形變成按鈕，控制按下後的動作，如:按下後改變文字(需要將改變的文字用變數取代 Text 裡的文字)。  3. 發出聲音與音檔匯入: 可先在學習吧作業區內提供學生因檔下載，從 Swift Playground 左上角選單內插入平板上的音檔。	10 **分鐘** 30 **分鐘**	Visio Swift Playground



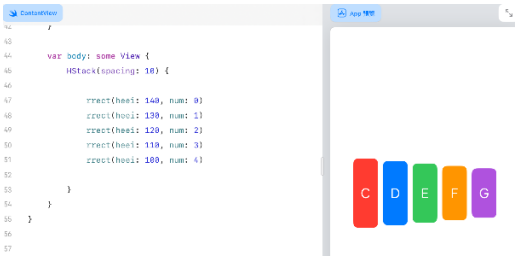
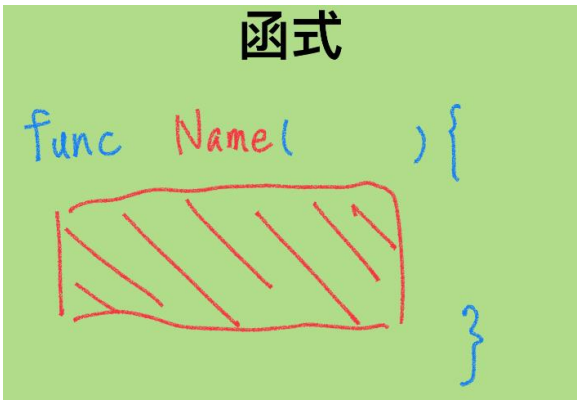
使用播放音檔功能需要先匯入模組，因此最前面要加上 import AVFoundation，再來設定一個播放器的變數 AVPlayer，透過 AVPlayer 播放 Bundle 內指定的音檔。接著按下動作裡面呼叫播放器模組。到此步驟按下矩形後，學生應該能聽到聲音。可以請學生依樣畫葫蘆做出第二個按鈕發出另一個聲音。

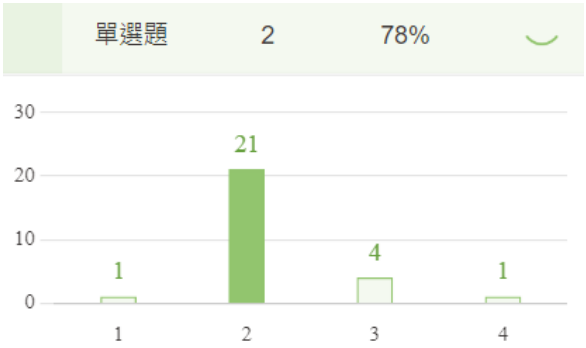
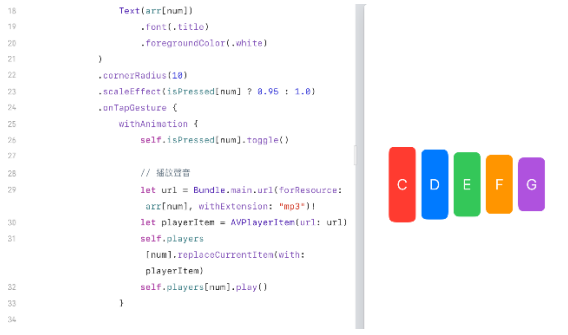
4. 排版(垂直與平行):
當有兩個以上按鍵時，原本的垂直排列想改成水平排列，可以將原本的 VStack 換成 HStack。



5. 修飾符:
文字或圖形可以用修飾符修改大小、顏色、樣式等等。讓學生自己調整，每個人做出來都可以獨一無二。

.fill(Color.black)修改矩形顏色 .cornerRadius(10)修改矩形圓角 .animation(.easeInOut(duration)) 點擊後縮放動畫  課程到此，每組應能跟著做出幾個圖形，點擊後能發出聲音。 各小組可能遇到的問題: 1. 使用複製貼上完成第二個矩形後，聲音檔案也換了，聲音卻出不來? 因為一個矩形按鍵需要獨立一個播放器變數，所以變數的地方也要新增。 2. 修飾符好多記不起來，但是又想讓畫面漂漂亮亮怎麼辦? 在學習吧放修飾符的語法讓學生參考，而且在SwiftUI裡只要打出前面幾個關鍵字，就會預測幾個可能是你要的語法供選擇。 3. 做了三四個按鍵後程式碼變得好長，很難看出這行程式是第幾個的程式，有辦法變短嗎? 所以接著要使用函數的方式來完成每一個按鈕，程式將大幅縮短。		
三、模組化的應用(II) (一)課前複習: 請同學到學習吧填寫測驗，看看上次的語法記得多少。老師可依照學習吧錯題分析進行語法複習。  	10 分鐘 25 分鐘	學習吧

(二)開啟上次的檔案，請同學仔細觀察。
詢問同學是否有地方重複出現?能否做成函數?
1. 建立圖形的程式。
2. 發出聲音的程式。
 	
(三)模組化:
教導學生 Swift 函數語法，接著請小組把程式依找先前繪製的流程圖，將能做成函數的地方加以模組化。
使用 func 函數名稱(參數設定){程式碼}來做副程式，將可以模組化的地方加以修改。記得函數要放在 var body:some View 之前。能使用參數設定的地方例如:每個矩形的長與寬不同，我們可以這樣設定函數
`struct ContentView: View {`
`@State private var player = AVPlayer()`
`func rrect(weei: CGFloat,heei: CGFloat) -> some View{`
`Rectangle().frame(width: weei, height: heei)`
`}`
`var body: some View {`
`VStack {`
`rrect(weei: 50.0, heei: 50.0)`
`}`
`}`
`}`
紅線、綠色部分定義參數，再到主程式裡面呼叫函數(黃線部分)。
 
 	
使用函數設定按鈕後整個程式簡潔很多。
 
 	
	10
	分鐘
		平板
		Swift Playground

<table>
<tr><td colspan="2">(四)成果分享與作業繳交:
小組討論後，分享程式模組化後的心得，感受到的優點。老師可示範當某些地方需要做修改，模組化後的方便性(用參數來修改形狀、顏色、大小…等)。</td><td></td><td>平板
Swift Playground
學習吧</td></tr>
<tr><td rowspan="4">教學成果</td><td> 
</td><td colspan="2"> 
</td></tr>
<tr><td>說明:學習吧課程安排</td><td colspan="2">說明:學習吧錯題檢討</td></tr>
<tr><td> 
</td><td colspan="2"> 
</td></tr>
<tr><td>說明:完成作品</td><td colspan="2">說明:用 Playground 練習函數，有預測語法，不用擔心單字拼錯</td></tr>
<tr><td>教學心得與省思</td><td colspan="3">學生對於文字式程式語言感到較陌生，因此進度會較緩慢，此時預留給學生操作的時間要比使用 Scratch 來的較久，讓學生分組進行討論也是考量到他們可以討論互助，用 Swift Playground 來學習文字式的程式語言會有語法預測，可以大幅降低拼錯單字的機會。如果將一些修飾語法(字形、大小、顏色…等)整理成表，放置學習平台供學生參考，學生其實很樂意去嘗試，他們都希望自己的版面要跟別人不一樣。最後將函數套入程式後，他們會更有感覺原本一長串的程式，突然變得很乾淨簡潔，不用上下滑來滑去</td></tr>
</table>

	找不到要修改的地方。這樣一來就達成很抽象的模組化概念，讓他們在修改程式的當下就能理解了。
參考資料	Michael Swift Playground 課程講義
附錄	**模組化的概念題目：** 1. 模組化的主要目的是？ ○ 提高程式碼行數 ○ 減少軟體的可擴展性 ○ 降低程式碼重用性 ○ 提高軟體的可維護性 2. 模組化可以把大問題拆成小問題 ○ O（是的） ○ X（不是的） 3. 整個程式不做模組化程式碼就不能運作 ○ O（是的） ○ X（不是的） 4. 模組化優點是？ ○ 主程式更容易閱讀 ○ 程式碼長度可以更長 ○ 占用更多記憶體 ○ 讓駭客更難入侵 5. 可以如何找到程式中可模組化的地方？ ○ 找最長的那串程式碼一定可以模組化 ○ 程式裡面一直出錯的地方 ○ 程式裡面一直重複出現的地方 ○ 程式裡面很多括號的地方 **語法知多少題目：** 1. 圖形想要水平排列應該使用哪個語法包起來? ○ A. ZStack ○ B. HStack ○ C. VStack ○ D. XSpace 2. Text()用來…? ○ A. 顯示圖形 ○ B. 間間隔 ○ C. 改變字體 ○ D. 顯示文字 3. 何謂修飾符? ○ A. 改變樣式 ○ B. 增加物件 ○ C. 用來增加變數 ○ D. 呼叫函數

4. import SwiftUI 用於?
    - A. 必要程式碼, 無意義
    - B. 匯入模組
    - C. 宣告變數
    - D. 建立函數
5. 所有程式不使用修飾符可行嗎?
    - A. 可以，只是會不大好看
    - B. 可以，但某些功能會故障
    - C. 不行，為必要程式碼
    - D. 不行，會增加程式卡盪機率
6. onTapGesture 功能?
    - A. 輸入文字
    - B. 新增圖形
    - C. 恢復為預設
    - D. 點擊後動作