

市本資訊科技領域教學綱要 國小組

程式、設計、思維

運算思維小組

Computational Thinking

- Seymour Papert 於 1980年首次使用這個辭彙。這位 MIT 教授是 LOGO 語言的共同發明人，並且是樂高機器人專案的合作開發人。致力於推動 K-12 的資訊科學教育，並認為運算思維是學習資訊科學的主要學習目標。
- 周以真 (Jeannette Marie Wing) 於 2006 年提議應將運算思維教給所有的學生，而非僅止於學習程式設計的學生，因為運算思維這種抽象思考能力，是解決問題最有效的方法，而問題解決學習法已經應用於所有的學科。

運算思維課程內涵

程式設計課程的學習內涵，應區分為運算思維與設計思維兩個部分，運算思維：是一種解決問題的系統化方法，包含四個解決問題的步驟：

1. 理解問題並嘗試將問題拆解成容易操作的步驟
(培養解構能力，以小御大)
2. 覺察問題出現的樣式，找出規律
(培養洞察力與歸納能力，一葉知秋)
3. 將問題虛擬化並塑造成數據模型
(運用數位素養與學科知識來表徵問題，深入淺出)
4. 設計演算法解決問題
(找出系統化且通用的方法解決相同類型問題，見招拆招)

以小御大

培養解構能力：

- 將數據縮小，讓問題變容易。
- 找出問題的所有變因，並逐一想辦法控制，讓問題變單純。
- 運用演繹邏輯思考，依問題的先後順序細步化，每次只解決一小部份。將問題程序化。
- 運用歸納邏輯思考，從其他領域的類似問題導入解法。旁敲側擊法。

一葉知秋

培養洞察力與歸納能力：

- 觀察問題出現的時機點是否有規律性。
- 觀察伴隨現象或數據是否有規律性。
- 觀察處理程序是否有規律性。
- 觀察規律性是否有遞移或轉換現象，找出通用法則。

深入淺出

運用數位素養與學科知識來表徵問題：

- 使用圖像或圖表來表徵問題(例如：捷運路線圖、心智圖)
- 使用符號來表徵問題(例如： $K=(P+I)*S$) 知識=(內在經驗+外在資訊)X 分享
- 選擇適當的資料結構來建立數學模型(例如：用陣列表示迷宮)
- 運用資料流程圖，解釋問題的成因(例如：電網效率分析)
- 運用程序流程圖，構想解決問題的程序(例如：SOP)

見招拆招

找出系統化且通用的方法解決相同類型問題：

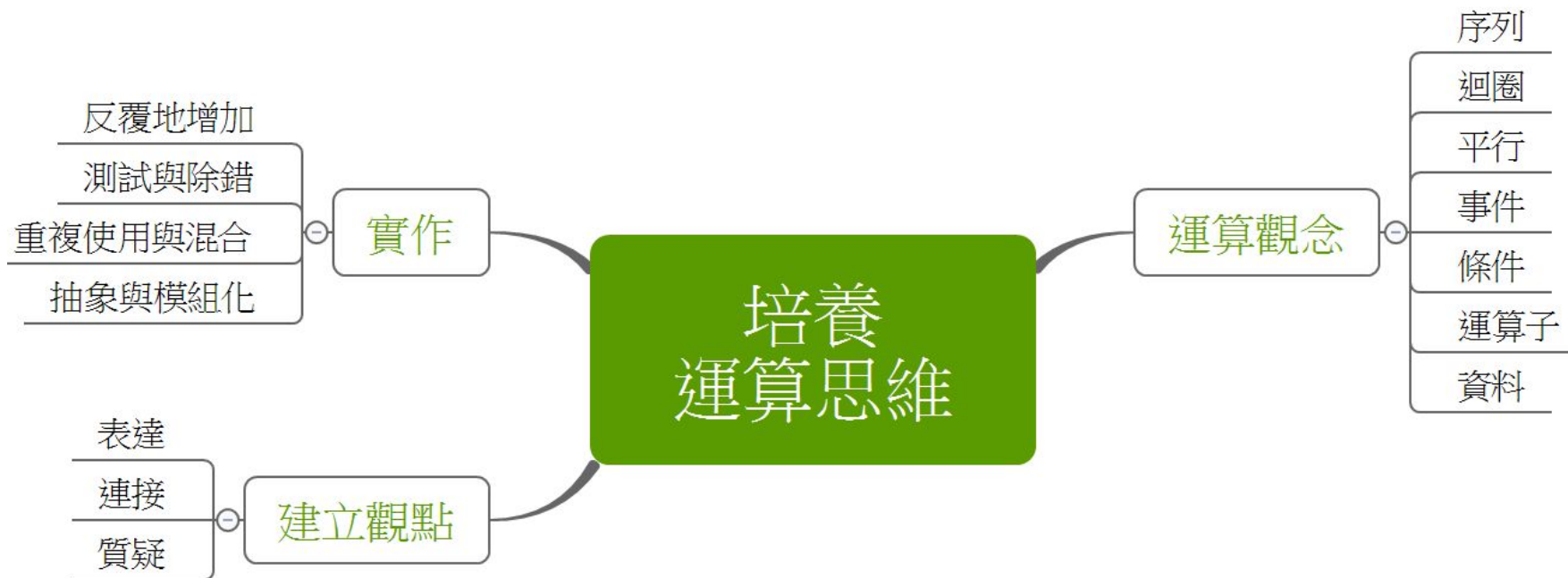
- 設計處理程序，能根據數據變化提供預測或判讀資訊。
- 檢查處理程序是否能處理例外情形。
- 將不同問題解決流程整合成一體。
- 將處理程序模組化，以便反覆利用。

運算思維的先備能力

- 閱讀理解能力(語文)
- 符號運算能力(數學)
- 邏輯組織與分析能力(哲學)
- 數位素養(非必備, 資訊)

學生須具備以上基本能力, 才能掌握運算思維, 美國教育界普遍認為國小四年級就可以在學習情境中運用運算思維。四年級以下可以先培養學習興趣並練習單項能力, 例如: 樣式識別。

皮亞傑認知發展論認為7~11歲為具體運思期, 11~16歲為形式運思期。MIT早期教育發展中心研究發現孩童從2歲可以辨識顏色後, 就具備了基本的邏輯推理能力, 例如: 分類、找相似、找不同。



什麼是設計思維？

- 設計的設計
- 關於設計的思考
- 以人類思考模式為基礎的設計（而非以電腦運算的模式）
- 基於人性的創意發想

為什麼要強調設計思維？

- 程式設計並非只是寫程式（Coding），也要學習設計原理和方法（Programming）
- 掌握運算思維幫助我們解決問題，掌握設計思維則幫助我們落實創意
- 創客 + 設計思維 = 文創產業

設計思維課程內涵

程式設計課程的學習內涵，至少應包含運算思維與設計思維，設計思維：是一種創意發想的系統化方法，包含四個執行專案的步驟：

- 了解並分析使用對象的需求
(培養敏察能力, 將心比心)
- 透過設計提供感動的使用者經驗
(培養美感與表達能力, 人因工程)
- 測試與除錯, 展示與溝通, 並持續改善
(培養溝通技巧與負責的態度, 誠意行銷)
- 從使用者的回饋進行評估與決策
(培養社會責任、永續環保、價值取向等信念, 創造價值)

教師的任務

- 興趣驅動學習，學習帶來應用，應用引發創造
- 任務一：教師應著眼於如何誘使學生對寫程式產生濃厚的興趣
- 任務二：教師應佈置應用的情境，提供待解決的問題
- 任務三：教師應允許甚至鼓勵學生的自由表現，並悅納學生的創意

學習目標

- 程式設計的學習，重點在設計方法的學習，而非程式語言的學習
- 運算思維可以遷移，而程式語言總會消失
- 程式設計學習的目標，應該是透過寫程式，學習如何透過設計去解決問題，然後把解決問題的經驗內化成運算思維

教學設計原則

- 運算思維是拿來用的，不是拿來教的
- 教師應設計學習活動讓學生練習使用運算思維，而不要去解釋它
- 教師應運用運算思維於課程活動設計中，將複雜問題拆解成簡單的討論程序



- **程式神秘箱**

學生透過作品的執行體驗，在看不到程式碼的情形下，預測程式內容並試做驗證。

- **大家來找碴**

老師提供有邏輯錯誤的程式碼，並說明程式的目的，學生必須發現錯誤並嘗試修正，讓程式可以如預期運作。

- **祕技與外掛**

學生從現成的範例遊戲出發，透過改變難度、修改操作方式、增加關卡 ... 等方式設計成更好玩的遊戲。

- **異想天開**

就老師給定的主題，學生自由發想故事情節，並設計成舞台劇動畫。

- **黑客松**

在限制規則內學生進行分組競賽，限制規則可包含：時間限制、積木限制、角色限制、輸入裝置限制 ... 等。

教學綱要

學習階段	編號	學習內容	建議實施節數
低年級	TP-I-1	序列式動畫設計	4
	TP-I-2	使用案例圖	4
中年級	TP-II-1	迷宮解題	4
	TP-II-2	物件導向概念	2
	TP-II-3	連鎖式動畫設計	8
	TP-II-4	腦力激盪會議	2
	TP-II-5	3D建模浮雕設計	4

教學綱要

學習階段	編號	學習內容	建議實施節數
高年級	TP-III-1	中文虛擬碼與流程圖	4
	TP-III-2	隨機數演算法	8
	TP-III-3	海龜作圖	4
	TP-III-4	基本環境感測	4
	TP-III-5	互動式遊戲設計	8
	TP-III-6	3D建模立體設計	4
	TP-III-7	背景故事設計及表達	-

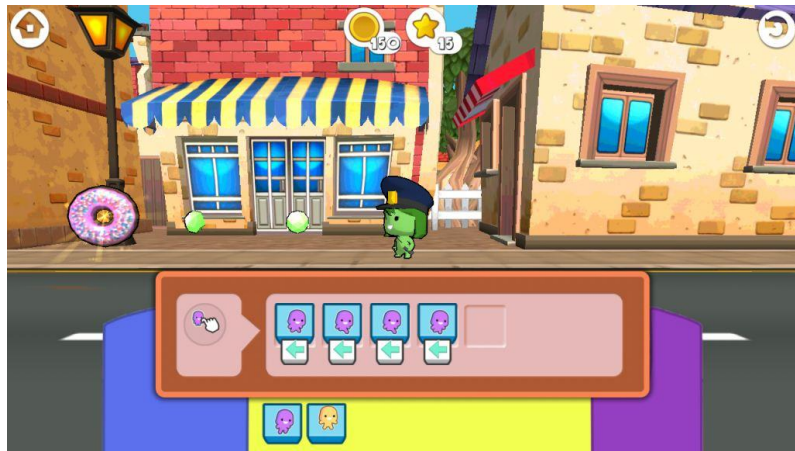
課程規劃

學習內容: 序列式動畫設計

教學資源: 紙卡、平板(可使用之 APP: Scratch Jr.、Daisy the Dino、The Foos...)

課程規劃:

此課程內容可以使用平板或紙卡進行(可參考 code.org 紙本教材或自行設計), 學生透過前進、後退、左轉、右轉、跳躍.....等簡單指令(不帶參數的指令), 串接成序列, 以進行解題或動畫設計, 剛開始先使用單一角色, 若學生程度較好, 可以導入多個角色, 學習多線式動畫設計, 例如: 舞會動畫設計。



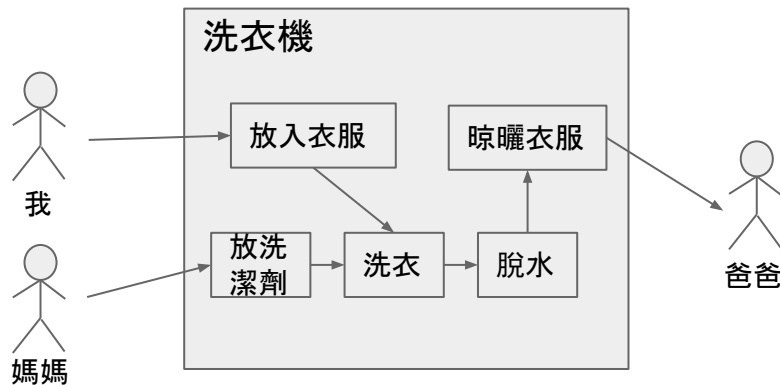
課程規劃

學習內容: 使用案例圖

教學資源: 紙筆、平板(心智圖軟體)或電腦(自由軟體 Dia、雲端服務 Cacao)

課程規劃:

此課程可使用UML軟體(如: Dia、Cacco...)或紙本手繪方式進行, 學生先練習將生活中常用的科技產品及設備, 繪製成使用案例圖, 藉以認識人機介面設計中, 參與者、動作、處理及回饋等要素。發展活動可以請學生設計幻想中的科技產品, 藉由使用案例圖表達該產品的功能與使用的便利性。



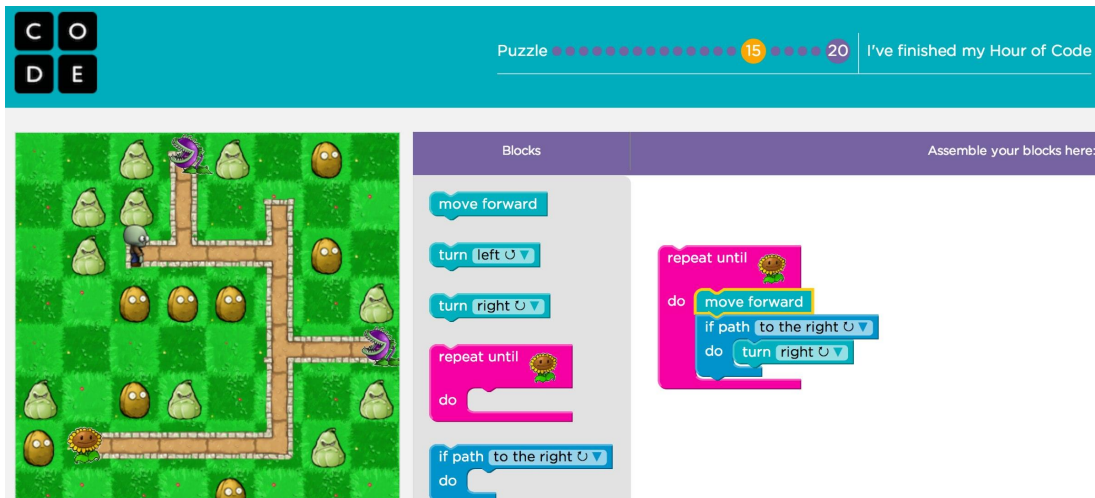
課程規劃

學習內容：迷宮解題

教學資源：學習網站(code.org、www.playcodemonkey.com、blockly-games.appspot.com)

課程規劃：

透過線上學習資源的關卡設計，學生運用運算思維進行迷宮解題，包含：拆解問題、找出重複樣式、設計演算法。程式語言層面的學習內容包含：敘述語法、條件判斷、重複結構、複合式條件判斷、分支結構 ... 等。學生循序漸進學習這些語法，並用來組合出能夠解題的演算法。



The screenshot displays a coding game interface. At the top, there is a teal header with a 'CODE' logo on the left and a progress indicator on the right showing 'Puzzle' with 15 out of 20 stars and the text 'I've finished my Hour of Code'. Below the header is a green maze with a path leading to a goal. The maze contains various obstacles like purple worms and brown eggs. On the right side, there is a 'Blocks' panel with a list of available blocks: 'move forward', 'turn left', 'turn right', 'repeat until', and 'if path to the right'. The 'Assemble your blocks here:' area shows a script being built, consisting of a 'repeat until' block containing a 'do' block with 'move forward', 'if path to the right' (with a dropdown arrow), and 'do' block with 'turn right' (with a dropdown arrow).

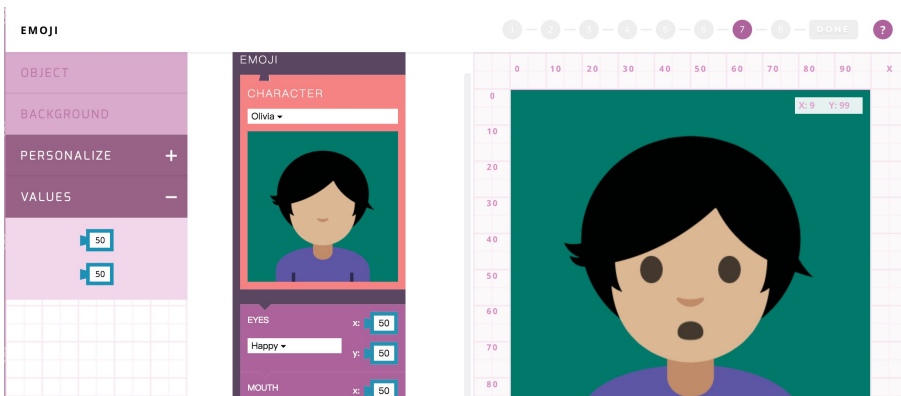
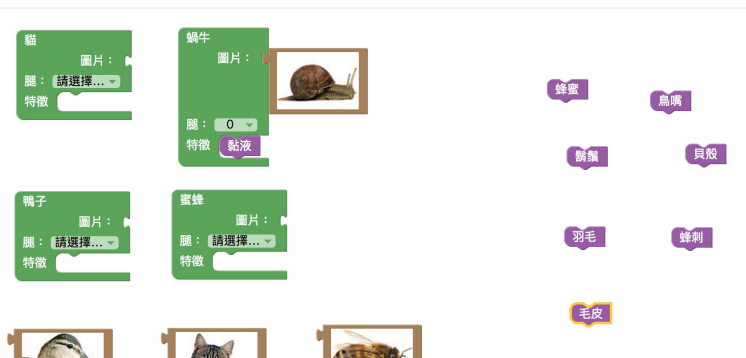
課程規劃

學習內容：物件導向概念

教學資源：電腦與網路 (Scratch)、學習網站 (www.madewithcode.com/projects/emojify、blockly-games.appspot.com)

課程規劃：

透過線上學習資源 (blockly...) 的物件導向分析操作，學生練習將操作對象建構成抽象物件，找出該對象的屬性及屬性質域，此練習將會學習到運算思維中的樣式識別、抽象化。同時也會練習到，如何將物件導向分析用於問題解決程序的規劃。在小學階段，物件導向概念主要教授內容為物件封裝與繼承，在此課程中所稱物件乃指類別，而視覺化語言中之物件乃指實體。物件導向為視覺化語言的隱含概念，對於視覺化語言的學習有很大的幫助。



課程規劃

學習內容：連鎖式動畫設計

教學資源：電腦與網路 (Scratch)

課程規劃：

此課程可以使用 Scratch 等視覺化程式語言來進行，學生在第一階段已經學習過如何運用簡單指令來設計動畫，在這個階段中將學習透過鍵盤或滑鼠產生之事件，與動畫角色進行互動。在發展活動中，學生可以練習如何運用碰撞偵測，啟動連鎖式動畫序列，例如：踢足球、骨牌效應、舞台劇 ... 等，藉此學習事件驅動的意涵以及多執行緒的分配與合併等重要概念。這些概念屬於潛在學習，不應直接提及或講解。程式語言層面的學習內容同 TP-II-1。



課程規劃

學習內容:腦力激盪會議

教學資源:白板和色筆、電腦(協作軟體或網站)

課程規劃:

腦力激盪會議於協同學習過程中會經常使用，但學生對於腦力激盪會議進行方式並不熟悉。老師可以透過討論主題的設計，例如：電流急急棒關卡設計，引導學生練習腦力激盪會議的進行方式，此課程可以選擇使用心智圖軟體(電腦或平板)來進行，也可以使用傳統白板和色筆來進行。會議主持人將標題寫在白板中央，參與者每人選一種色筆(顏色不要重複)以心智圖的形式，寫出自己的想法。所有內容都不塗改，而是以刪除線標示。教師應展示每個小組的討論結果，並引導學生從顏色分辨每個人在討論中的貢獻。



課程規劃

學習內容: 3D建模浮雕設計

教學資源: 3D建模軟體或網站(www.tinkercad.com)

課程規劃:

3D建模依照難易度區分, 約略可分為: 浮雕、立體單一形體、立體組合形體。此階段應學習如何建立2.5D浮雕模型, 例如: 名牌、杯墊、鑰匙圈 ...等, 表現技法可以是陰刻或陽刻。若有輸出設備, 應讓學生輸出作品以獲得動手做的成就感。



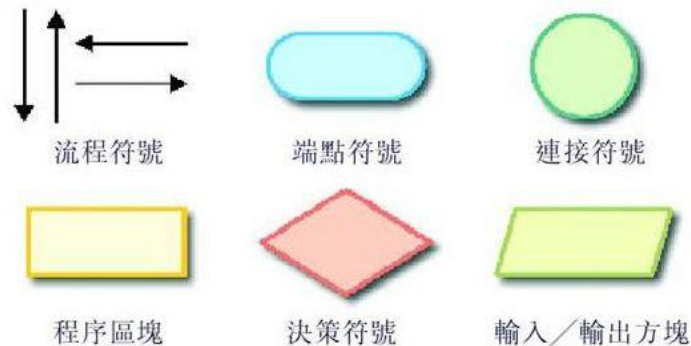
課程規劃

學習內容: 中文虛擬碼與流程圖

教學資源: 紙筆、平板(心智圖軟體)或電腦(自由軟體 Dia、雲端服務 Cacao)

課程規劃:

學習程式設計的過程中，經常需要閱讀別人的程式碼，在這個課程裡，先讓學生練習用中文敘述別人的程式邏輯，在沒有圖像輔助的情形下是否能讓全班同學聽懂。在發展活動中，老師運用教學鷹架 - 程式神秘箱，讓學生練習用中文預測並表達程式邏輯，然後根據中文敘述繪製成流程圖(用中文虛擬碼取代程式碼)。



課程規劃

學習內容：隨機數演算法

教學資源：電腦與網路 (Scratch)

課程規劃：

隨機數衍生之演算法運用範圍很廣泛，是學生在進行遊戲設計之前，必須先學會的概念。最簡單的隨機數演算法可用於生態模擬，例如：魚的游動、醉漢走路 ...等，當學生練習完生態模擬的案例後，接下來可以進行抽籤機的專題製作，由簡到難分別為：連續號碼抽籤、非連續號碼抽籤、不重複抽籤。學生在課程中將學會如何運用變數、陣列，配合隨機數完成演算法的設計。此單元相關運算思維包含：拆解問題、抽象化、數學建模、設計演算法。資料結構層面的學習 內容包含：串列索引與 值域之耦合關係、串列 (list) 操作..等。



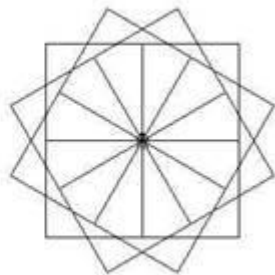
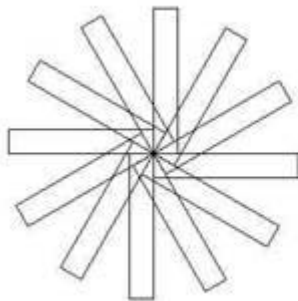
課程規劃

學習內容：海龜作圖

教學資源：電腦與網路 (Scratch、Blockly)

課程規劃：

海龜作圖起源於LOGO語言，是學生學習數學建模最好的方法，從畫直線開始，然後畫多邊形、圓形、螺旋線、曲線、碎形。在這個階段應讓學生練習使用多邊形、圓形、螺旋線進行圖案的組合和設計。曲線與碎形可以視學生程度決定是否安排到課程中。學生運用運算思維進行海龜作圖，包含：拆解問題、找出重複樣式、設計演算法。程式語言層面的學習內容包含：巢狀迴圈、程序模組化、複合式條件判斷、分支結構 ...等。



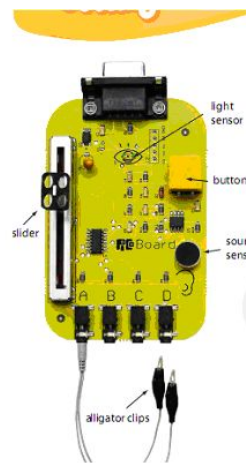
課程規劃

學習內容: 基本環境感測

教學資源: 電腦與網路 (Scratch) + 外接電路板 (picoboard、樹莓派、arduino...)

課程規劃:

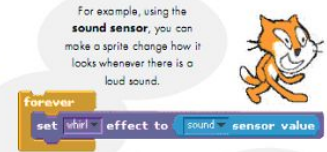
使用外接電路板所提供的感測器進程式設計, 例如: picoboard、樹莓派、arduino...等。教師應根據感測器種類來設計課程主題, 例如: 電阻感測可用來設計電流急急棒遊戲, 聲音感測可用來設計聲控遊戲 ... 等。在沒有採購額外電路板的情形下, 可以使用 Scratch 2.0線上版提供的聲音感測和影像感測來設計體感遊戲。



Connect real-world sensors to your Scratch projects


The **PicoBoard** enables your Scratch (scratch.mit.edu) projects to sense – and respond to – things going on in the world outside your computer.

For example, using the **sound sensor**, you can make a sprite change how it looks whenever there is a loud sound.



Or, using the readings from the PicoBoard's **light sensor**, you can program a Scratch sprite to hop up and down whenever the sensor detects a passing shadow.

You can use the **slider** and **button** to control a character in a video game.



課程規劃

學習內容：互動式遊戲設計

教學資源：電腦與網路 (Scratch)

課程規劃：

此課程的學習內容，應結合TP-III-7學習內容，以建立遊戲設計概念為目標。教師可以運用程式設計教學鷹架，採用多元的教學方法，根據學生能力選擇適合的題材進行遊戲設計教學。例如：撈金魚、打地鼠、青蛙過街、走迷宮、敲磚塊...等。此單元相關運算思維包含：拆解問題、抽象化、數學建模、設計演算法。程式語言層面的學習內容包含：複雜演算式、平行處理、啟發式演算法...等。



課程規劃

學習內容: 3D建模立體設計

教學資源: 3D建模軟體或網站(www.tinkercad.com)

課程規劃:

3D立體單一形體的複雜度不一，從簡單的柱體、錐體、球體到螺旋體，設計的難度會逐步提升。建議此階段仍以柱體、錐體為主，輔以其他幾何形體的組合，例如：紀念碑、獎盃、機器人公仔 ...等，學生在此課程中練習幾何形體的組合、挖空等技法去形塑自己的創意作品。若有輸出設備，應讓學生輸出作品以獲得動手做的成就感。



課程規劃

學習內容: 背景故事設計及表達

教學資源:

課程規劃:

無論是遊戲設計或是 3D 模型的設計，設計者應該提供背景故事，以此作為創意發想的起點，同時背景故事也可以提供一些設計要素到作品中，學生在發表作品時應該清楚表達作品設計的理念。以打地鼠遊戲設計為例，遊戲開始前的劇情介紹，敘述外星人入侵地球，破壞地表的意圖以說明為何要打地鼠。作為外星人偽裝的地鼠，會具備一些外星人超能力，例如：瞬間移動 ... 等，以此發想新的玩法。此課程內容應結合其它課程一起進行，可以作為腦力激盪會議的討論主題，也可以作為作品發表報告內容的一部份。

建議與指教