



視覺化程式設計語言

Visual Programming Language (VPL)

運算思維與程式設計學習

國小階段

- 資訊科技之體驗與應用
- 目標導向運算思維學習
 - 程式設計語言 (VPL) 學習

國中階段

- 以運算思維與資訊科技解決問題
- 問題導向運算思維學習
- 依課程內容選擇VPL或是non-VPL進行教學

高中階段

- 了解運算思維原理，並能進一步整合應用
- 創作導向運算思維實踐
- Java/C/C++/Python
- Arduino, App Inventor



視覺化程式設計語言

圖型化元素
代替文字



視覺表達



淺顯易懂



- 程式初學者
- 先備知識不足

視覺化程式語言

Scratch

Kodu

Blockly

App
Inventor

S4A

遊戲設計

轉換
文字程式

App
專題製作

程式設計

程式設計基本概念

問題解決與規劃
資訊科技應用



	Scratch	Kodu	Blockly	App Inventor
開發者	MIT	Microsoft	Google	MIT
中文介面	✓	✓	✓	✓
操作環境	網頁/單機版	單機版	網頁	網頁/單機版
介面	2D	3D	2D	2D
線上發佈	✓			✓
轉換文字程式			✓	
教學導向	物聯網/ 專題製作	遊戲設計	問題解決	專題製作

視覺化程式設計語言

- Scratch

- 設計思維與介面
- Starter Projects
- 循序漸進學習Scratch

- Kodu

- 設計思維
- Maze Tutorial

- Blockly

- Blockly Games Project
- C Thinking
- Blockly Demos 程式碼

- App Inventor

- 設計思維與介面
- 打地鼠遊戲

- Scratch延伸操作：S4A

- Arduino 介面介紹
- S4A 介面介紹
- Demo- 眼明手快





Blockly



Blockly

Google發表

以網頁為基礎的
視覺化程式設計語言



支援中文介面

Open Source



Blockly

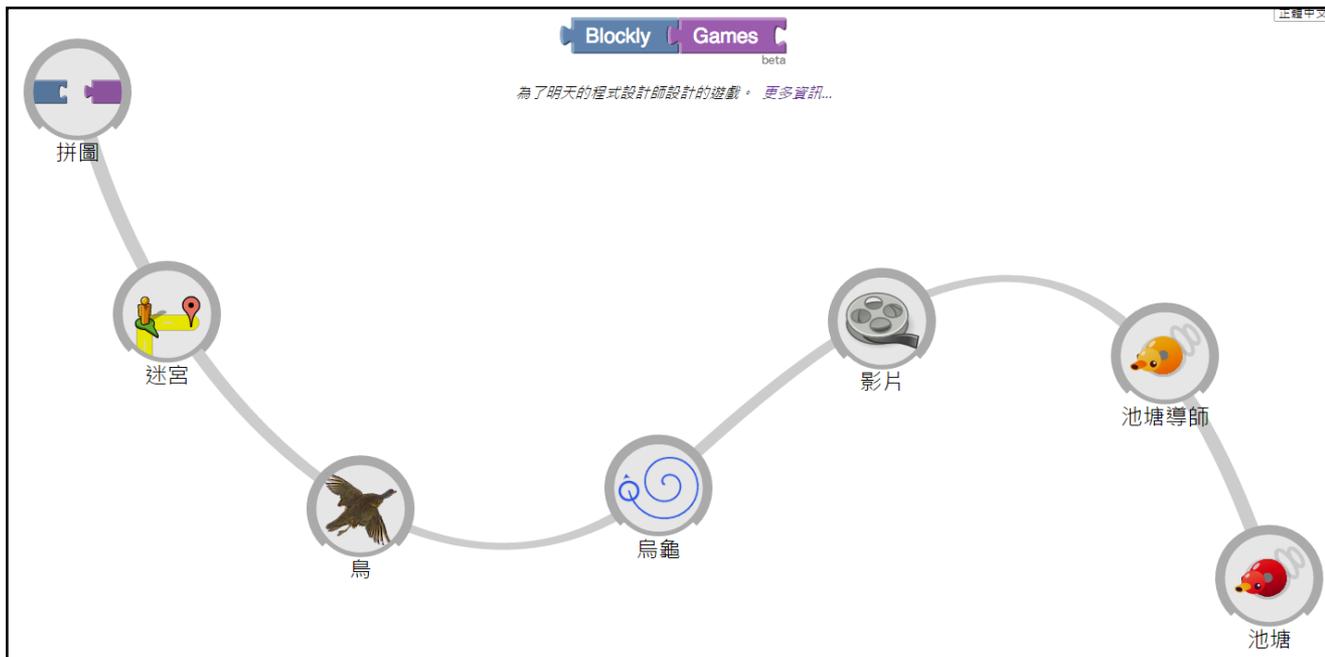
- Blockly Games Project
 - 利用動畫解決問題
- C Thinking
 - 跳脫動畫，進行問題解決
- Blockly Demos 程式碼
 - 跳脫動畫，進行問題解決



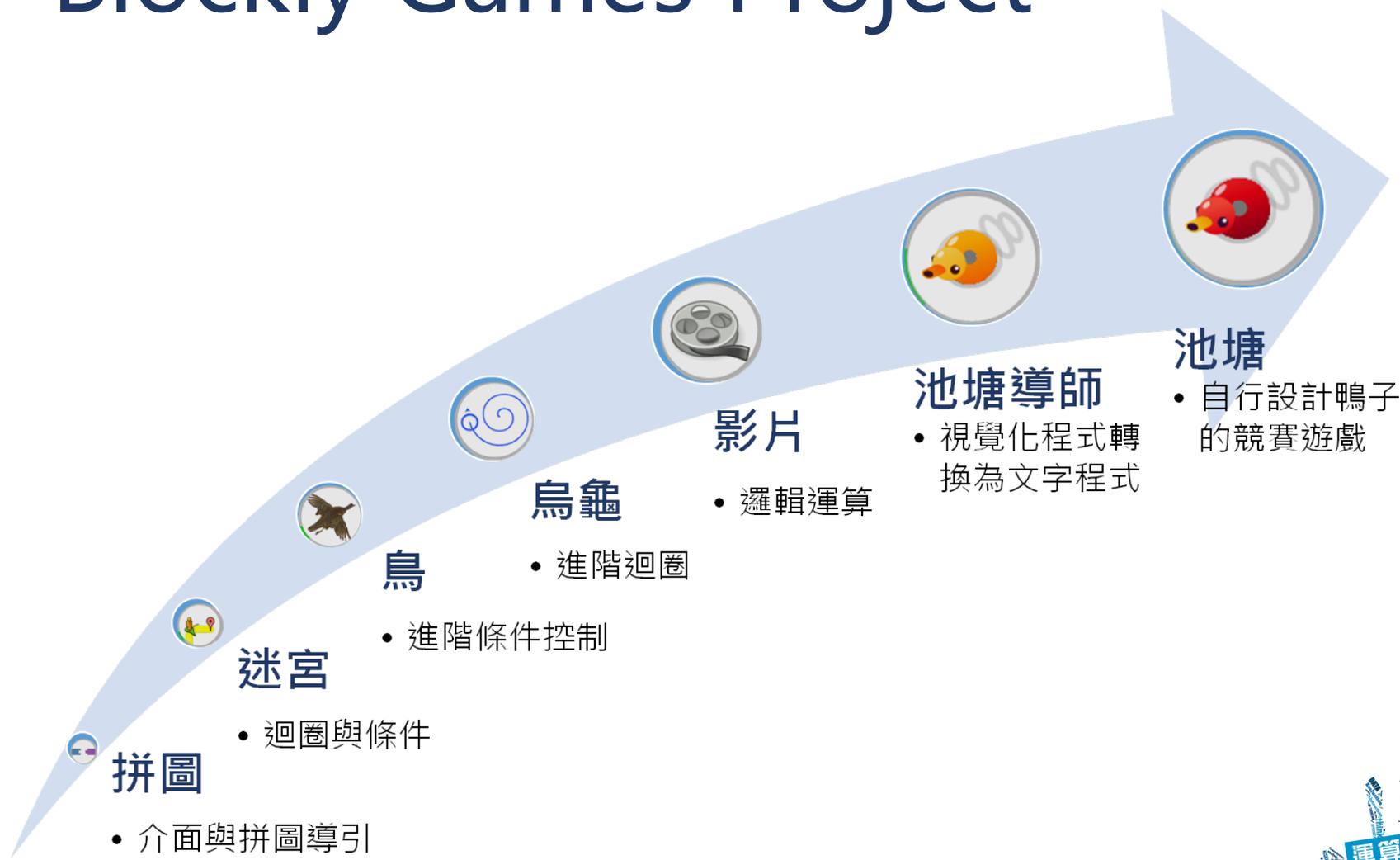
Blockly Games Project

- 利用動畫輔助學習

- <https://blockly-games.appspot.com/>



Blockly Games Project



C Thinking

- <http://ct2prog.csie.ntnu.edu.tw/>

- 程式設計與問題解決

- 跳脫動畫
- 循序漸進
- 學科整合

The screenshot shows the C Thinking website interface. At the top, there is a navigation bar with the logo 'C THINKING', the user name 'casey', and links for '回首頁', '學習檔案', and '登出'. Below the navigation bar is a grid of eight article cards. Each card features a yellow icon, a title, and a short paragraph of text.

Icon	Title	Text
Healthcare robot	健康管理小幫手	<穿戴裝置近幾年開始興起，智慧手錶、智慧手環等方便隨身攜帶的特性，吸引使用者透過這些裝置來觀察個人健康，並與行動裝置連結，記錄管理個人健康資訊。
Car	高速公路計程收費	高速公路電子收費系統（ETC）已經由收費站計次收費改為計算里程收費，但是費率內容除了分成大、小型車，還有免費里程和長途折扣優惠，讓很多人搞不懂計費方式。
Question mark	麥當勞都是為您？！	美味的漢堡、香噴噴的炸雞總讓人垂涎三尺，不過這相當於熱量炸彈的油炸食物，對於身材的保持可是極大的考驗，就讓我們先來了解並認識這些熱量炸彈吧...
Lightning bolt	電力補給站	在這人人一台智慧手機的時代，出門在外，行動電源也已成爲智慧型手機必備周邊之一了，市面上的行動電源五花八門，到底行動電源容量要怎麼挑選適合自己手機的呢？...
Paper airplane		
Person with 'Go!'		
Shopping bag		
Gavel		



C Thinking

循序漸深的題目練習

健康管理小幫手

近幾年來穿戴裝置逐漸興起，智慧手錶、智慧手環等方便隨身攜帶的特性，吸引著使用者透過這些裝置來觀察個人健康，並與行動裝置連結來記錄管理個人健康資訊。現在請根據穿戴裝置傳送進來的資訊，設計個人健康管理的小幫手。

衛生福利部國民健康署肥胖防治網

Q1

Q2

Q3

Q4

Q5

◀ 回上頁

可同步至學習檔案

「計步器」的功能是可以記錄你今天行走步數，請根據身高與步數來估算行走距離。

身高(公分) h
步數(步) steps
距離(公里) dist
 $dist = (h-100) * steps / 100 / 1000$

[輸入1]

h : 180

steps : 1000

[輸出1]

0.8

[輸入2]

h : 180

steps : 1000

[輸出1]

0.8

▶ 執行

📁 開啟舊檔

📄 下載至電腦

判斷
迴圈
運算
輸入/輸出
陣列
變數

```

set h to prompt for number with message “ 身高 ”
set steps to prompt for number with message “ 步數 ”
print (h - 100) * steps / 100 / 1000
  
```

英文程式方塊

☁ 同步至學習檔案

變數的值

h:180
steps:1000

輸出結果

0.8

多筆測資提供測試

健康管理小幫手

- 15分鐘體驗（完成越多關卡越好）

健康管理小幫手 Q1 Q2 Q3 Q4 Q5 < 回上頁

近幾年來穿戴裝置逐漸興起，智慧手錶、智慧手環等方便隨身攜帶的特性，吸引著使用者透過這些裝置來觀察個人健康，並與行動裝置連結來記錄管理個人健康資訊。現在請根據穿戴裝置傳送進來的資訊，設計個人健康管理的小幫手。

衛生福利部國民健康署肥胖防治網

「計步器」的功能是可以記錄你今天行走步數，請根據身高與步數來估算行走距離。

身高(公分) h
步數(步) steps
距離(公里) dist
 $dist = (h-100) * steps / 100 / 1000$

[輸入1]
h : 180
steps : 1000
[輸出1]
0.8

[輸入2]
h : 155
steps : 2500
[輸出2]
1.375

判斷
迴圈
運算
輸入/輸出
陣列
變數

變數的值

輸出結果

執行 開啟舊檔 下載至電腦 同步至學習檔案



Blockly Demos 程式碼

- <https://goo.gl/aXf1Ui>
- 程式設計與問題解決



[Blockly](#) > [Demos](#) > 程式碼

積木

JavaScript

Python

邏輯

迴圈

數學式

文字

列表

顏色

變量

流程

The image shows a Blockly script for calculating BMI. It starts with three variables: '身高' (Height), '體重' (Weight), and 'BMI', all initialized to 0. Then, it prompts the user to input height and weight. Finally, it calculates BMI using the formula $\text{BMI} = \frac{\text{Weight}}{\text{Height} \times \text{Height}}$ and outputs the result.

```

    賦值 身高 到 0
    賦值 體重 到 0
    賦值 BMI 到 0
    賦值 身高 到 輸入 數字 並顯示提示訊息 “ 請輸入身高(以公尺為單位) ”
    賦值 體重 到 輸入 數字 並顯示提示訊息 “ 請輸入體重(以公斤為單位) ”
    賦值 BMI 到 ( 體重 ÷ ( 身高 × 身高 ) )
    印出 建立字串使用 “ 你的BMI為: ” BMI
  
```

blockly-demo.appspot.com says:

透過此連結分享您的積木組：

<https://blockly-demo.appspot.com/static/demos/code/index.html?lang=zh-hant#874fi7>

OK

正體中文

GO

儲存積木組
並提供連結

學生可將網址繳交給教師進行程式批改



Blockly 參考資料

- 教學影片 **英**
 - Google Blockly Games - Solution All Level (Complete)
 - <https://www.youtube.com/watch?v=TG3H7xnHv-U>
- Blockly Games Project **中**
 - <https://blockly-games.appspot.com/>
- Blockly Demo 程式碼 **中**
 - <https://goo.gl/aXf1Ui>
- C Thinking **中**
 - <http://ct2prog.csie.ntnu.edu.tw>





Scratch



Scratch

MIT發展

可創造互動式
故事、動畫、遊戲

SCRATCH



支援
中文介面

編輯方式：
1. 瀏覽器
2. 離線版

可將開發程式
上傳至線上社
群分享



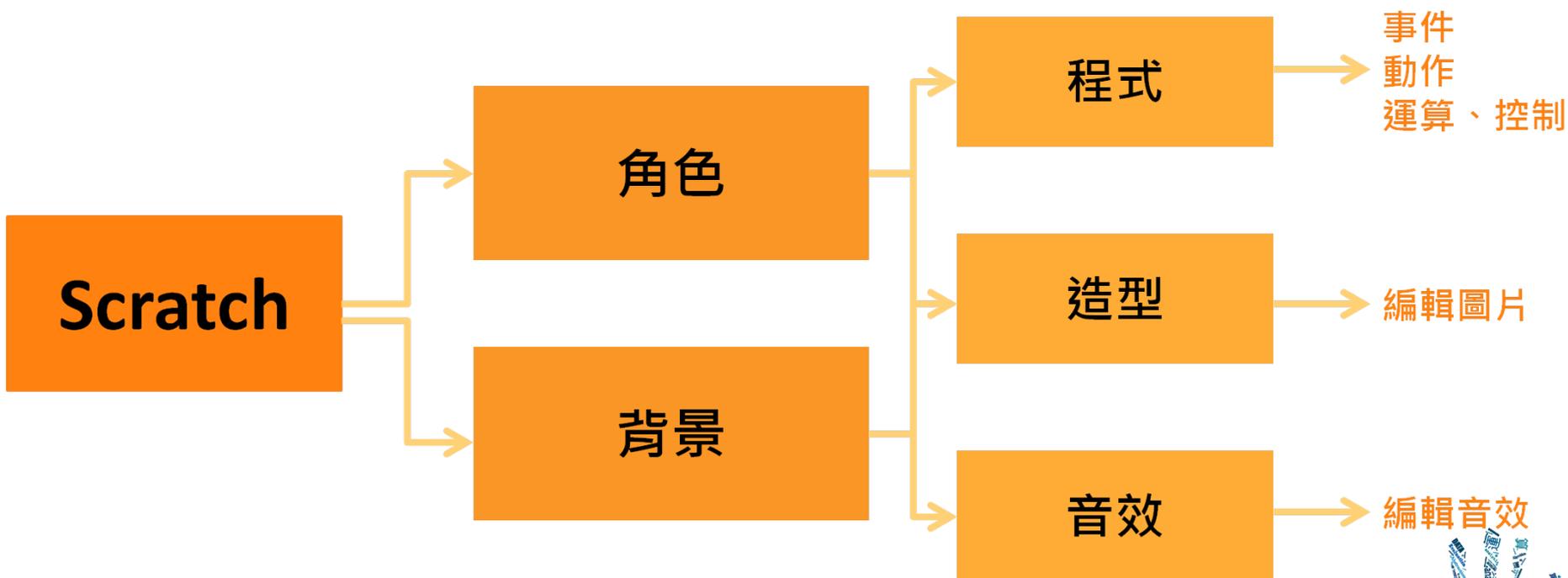
Scratch

- 設計思維與介面
- Starter Projects
- 循序漸進學習Scratch

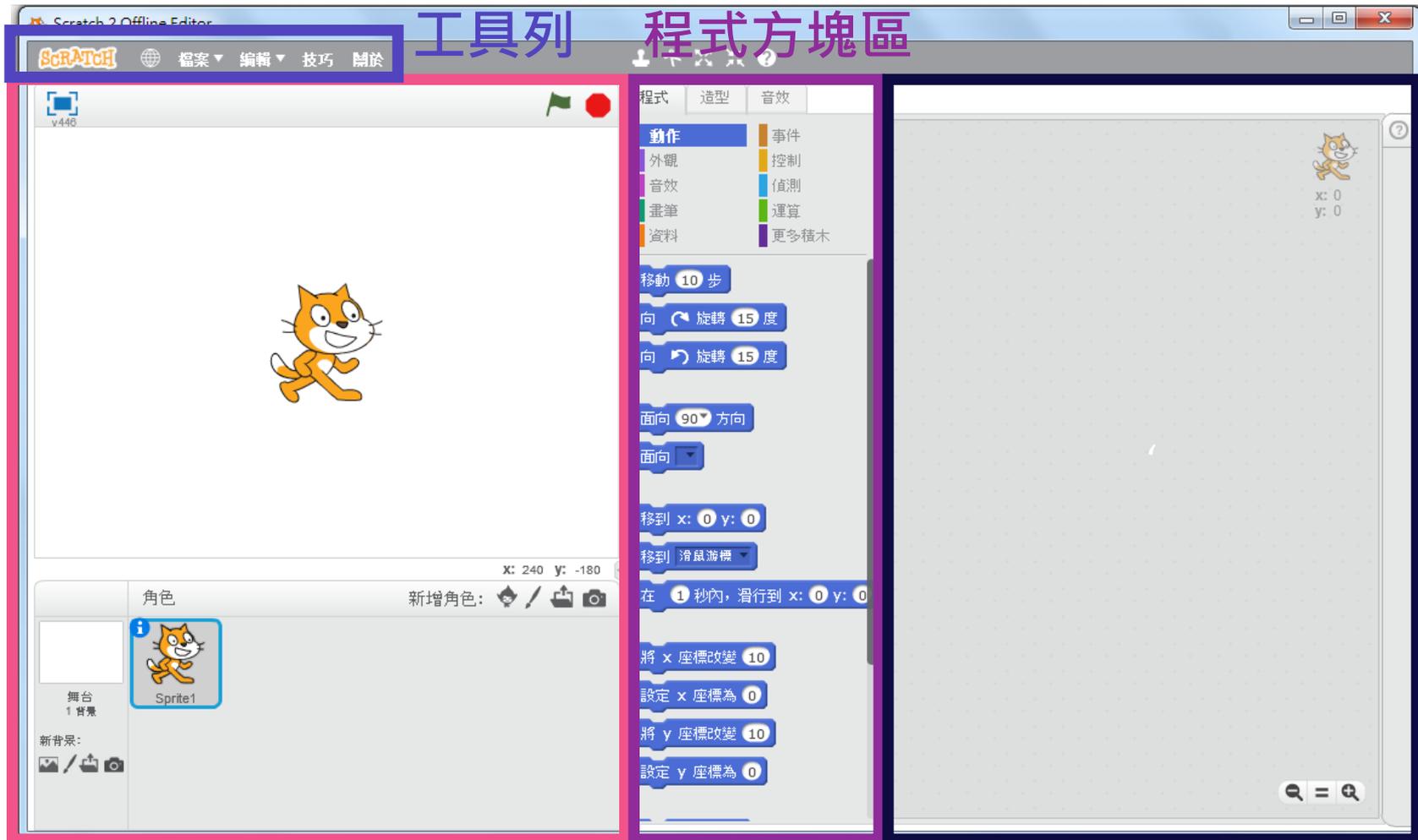


Scratch設計思維

- 建立不同的物件，並利用程式改變物件之表現



Scratch 介面

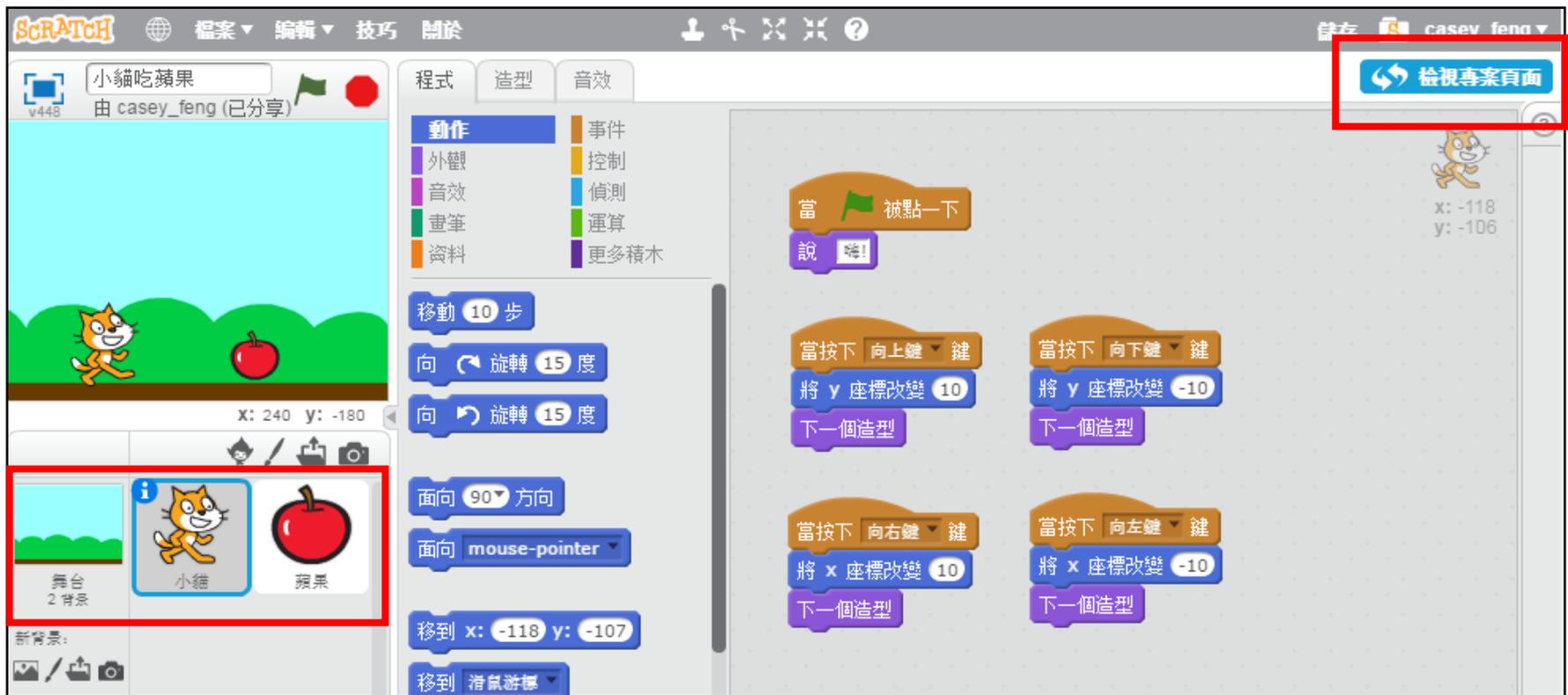


舞台編輯區

程式編輯區

Scratch程式

可將作品上傳至網頁

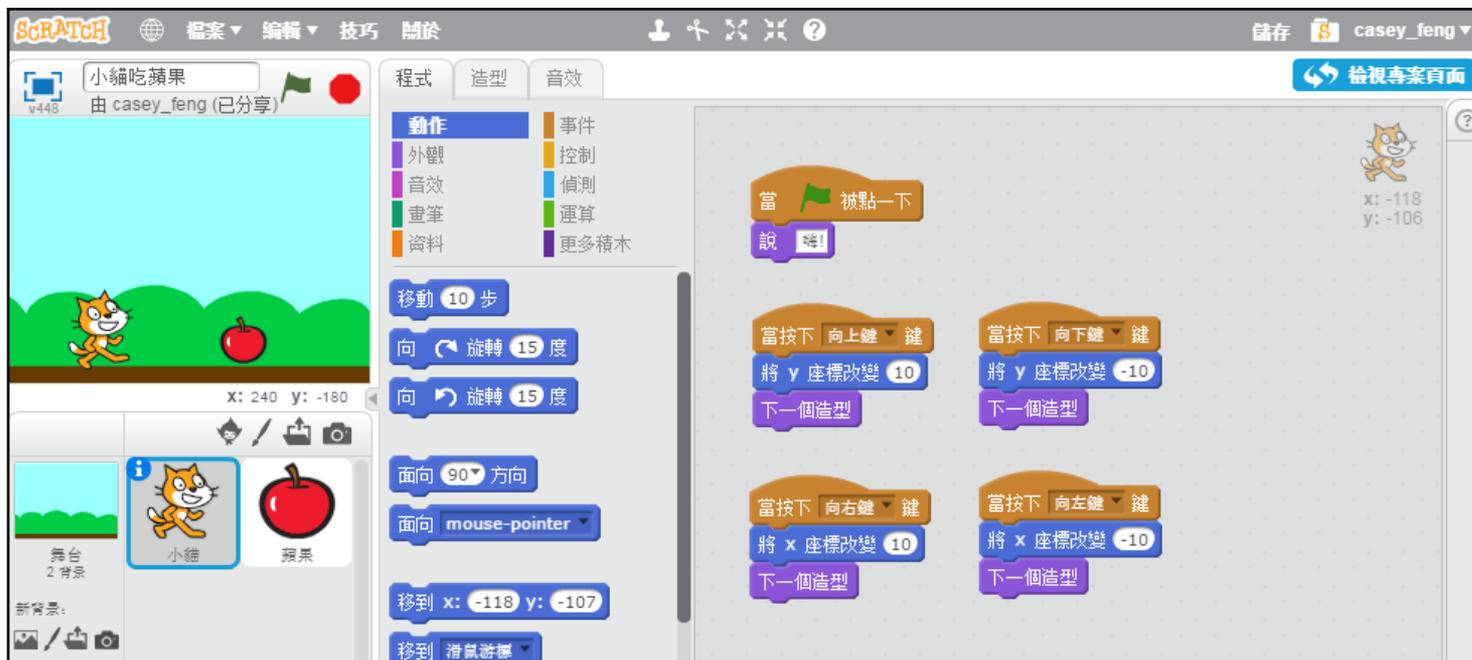


針對不同角色與背景進程式編輯



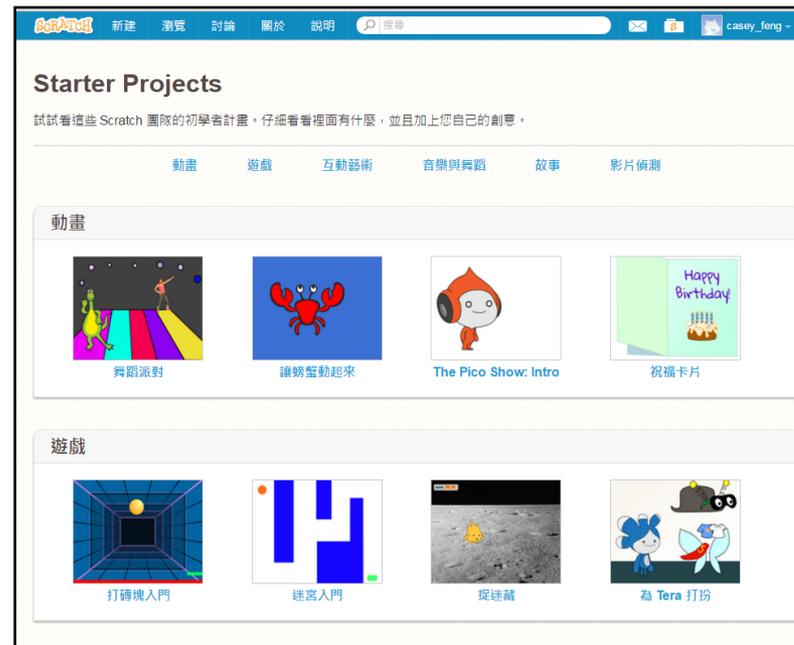
小貓吃蘋果

- 利用鍵盤(上下左右鍵)控制小貓
- 當小貓碰到蘋果時，蘋果會消失



Starter Projects

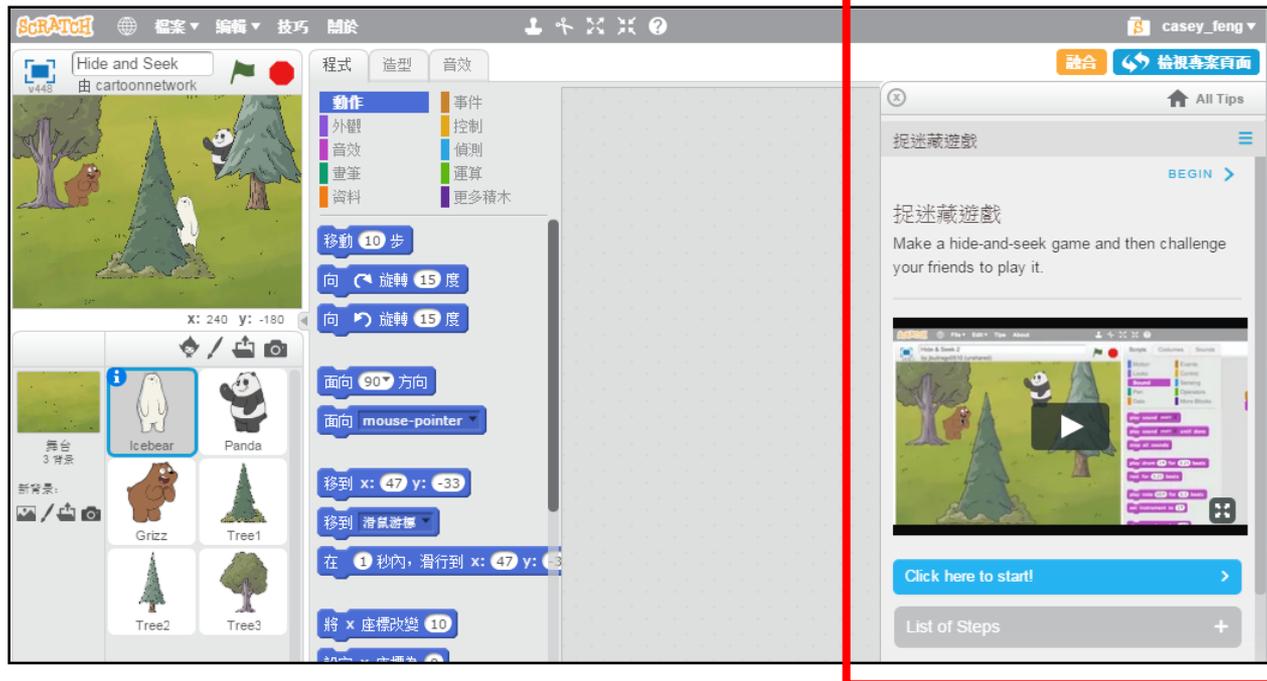
- https://scratch.mit.edu/starter_projects/
- 利用現有程式進行改寫



循序漸進學習Scratch

- <https://scratch.mit.edu/hoc>
 - 程式包裝
 - 循序引導

題目說明與
程式撰寫引導



參考資料 中

• 網站

- 均一教育平台
 - <http://www.junyiacademy.org/computer-science-a/isc/programming-scratch-1>
- Scratch中文學習網
 - <http://s4a.tw/>
- 1know
 - <http://1know.net/knowledge/db805d5d0723?v=list>
- Hour of Code
 - <https://scratch.mit.edu/hoc>
- Starter Projects
 - https://scratch.mit.edu/starter_projects/





Kodu



Kodu

微軟發展

可於電腦上或
Xbox執行



KODU

支援
中文介面

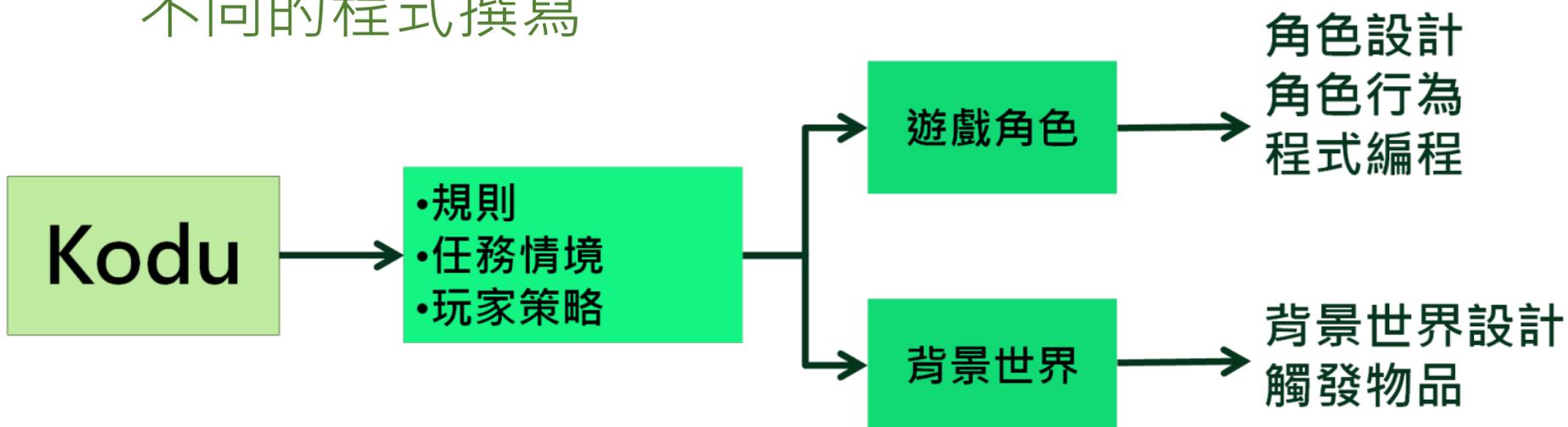
需安裝軟體
進行編輯

3D介面



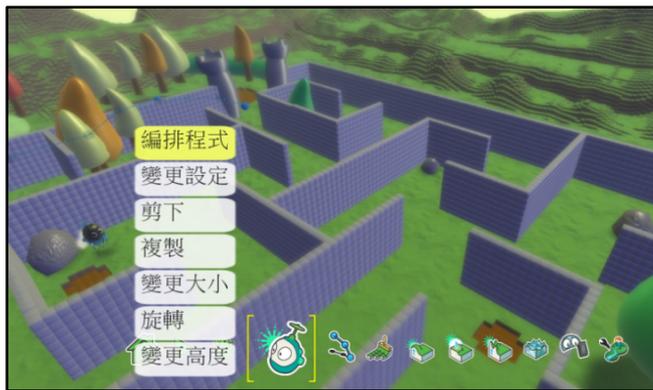
Kodu設計思維

- 給予遊戲情境並建立物件，針對物件與情境進行不同的程式撰寫



Maze Tutorial

- 藍色的Kodu在迷宮中迷路了，而白色的Kodu要來拯救他。



Kodu的3D介面



程式撰寫介面
(指令搭配圖案)

Maze Tutorial

- 操作白色的Kodu找到迷宮入口。
- 進入迷宮後，找到藍色的Kodu，然後帶他回家



參考資料

- 網站 **英**
 - <http://www.kodugamelab.com/>
- 教學影片 **英**
 - Kodu頻道
 - <https://www.youtube.com/user/KoduTeam>
- 中文社群 **中**
 - <https://www.facebook.com/groups/1570224519906656>
- 書籍 **中**
 - 微軟輔導手冊
 - <http://stream.kh.edu.tw/assets/documents/46/original/d41d8cd98f00b204e9800998ecf8427e.pdf>





App Inventor



App Inventor

MIT發展

只支援
Android系統
網站提供模擬器
提供測試

支援
NXT機器人



支援
中文介面

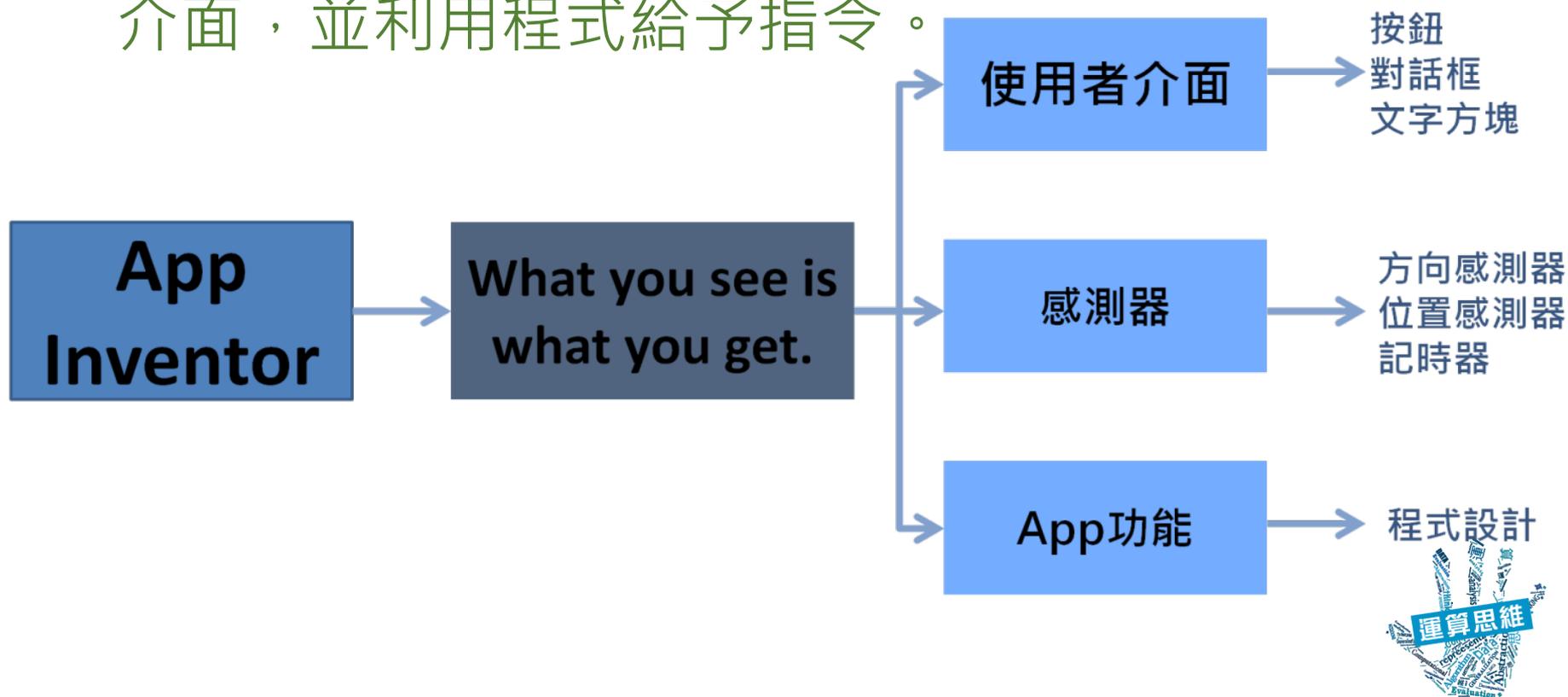
編輯方式：
1. 瀏覽器
2. 離線版

程式可上傳至
Google Play
商店供人下載



App Inventor設計思維

- 針對設計的App，需要什麼元件就將其放入App介面，並利用程式給予指令。



App Inventor設計介面

- 外觀設計的介面

The screenshot displays the App Inventor design interface for a project named "easy_mashmole". The interface is divided into four main sections:

- Component Palette (元件面板):** Located on the left, it lists various UI components such as Buttons, Image Sprites, Labels, and Clocks, categorized under "User Interface" (使用者界面) and "Screen Layout" (版面布局).
- Workbench (工作面板):** The central area showing a mobile device simulation. The screen displays a green header with the text "開始遊戲" (Start Game) and a Mashmole character in the center. Below the screen, there are controls for "Hidden Elements" (不可見元件) including Sound1, Clock1, Clock2, and Notifier1.
- Component List (元件清單):** Located on the right, it shows a tree view of the components on the screen, including Screen1, Label1, Button_START, Canvas1, ImageSprite1, Sound1, Clock1, Clock2, and Notifier1.
- Properties Panel (元件屬性):** On the far right, it shows the properties for the selected component (Screen1), including application name, background color, and animation settings.

Labels at the bottom of the interface identify the sections: "元件面板" (Component Palette), "工作面板" (Workbench), "元件清單" (Component List), and "元件屬性" (Properties Panel).

App Inventor 設計介面

- 程式設計的介面

The screenshot displays the App Inventor interface for a game named "easv mashmole". The interface is divided into three main sections:

- 程式方塊區 (Code Blocks Area):** Located on the left, it shows a list of available code blocks categorized by function (e.g., flow control, logic, math, text, clearing, color, numbers, and programs). A tree view on the right shows the current screen's components: Label1, Button_START, Canvas1, ImageSprite1, Sound1, Clock1, Clock2, and Notifier1.
- 工作面板：程式編輯區 (Workbench: Code Editing Area):** The central area where code is written. It features three event-driven code blocks:
 - 初始化全域變數 (score) 為 0:** A block to initialize the global variable 'score' to 0.
 - 當 Button_START 被點選:** A block triggered by the 'START' button click. It contains:
 - 執行 設 Clock1 啟用計時 為 true
 - 執行 設 Clock2 啟用計時 為 true
 - 當 ImageSprite1 被觸碰: A block triggered by the image sprite being touched. It contains:
 - 執行 呼叫 ImageSprite1 移動到指定位置 (x座標, y座標)
 - x座標 整數亂數從 0 到 319
 - y座標 整數亂數從 0 到 319
 - 呼叫 Sound1 播放
 - 設 global score 為 求 global score + 1
 - 設 Label1 文字 為 求 global score
 - 初始化全域變數 clock1_time 為 10
 - 當 Clock1 計時: A block triggered by Clock1's timer. It contains:
 - 執行 呼叫 ImageSprite1 移動到指定位置 (x座標, y座標)
 - x座標 整數亂數從 0 到 319
 - y座標 整數亂數從 0 到 319
 - 設 global clock1_time 為 求 global clock1_time - 1
 - 設 Button_START 文字 為 求 global clock1_time
- 當 Clock2 計時:** A block triggered by Clock2's timer. It contains:
 - 執行 設 Clock1 啟用計時 為 false
 - 執行 設 Clock2 啟用計時 為 false
 - 呼叫 ImageSprite1 移動到指定位置 (x座標, y座標)
 - x座標 135 = 320(螢幕寬高)/2 - 50(圖片寬度)
 - y座標 135
 - 呼叫 Notifier1 顯示消息對話框 (消息, 標題, 按鈕文字)
 - 消息 合併文字 "您的分數為" 求 global score "分"
 - 標題 "回合結束"
 - 按鈕文字 "OK"
 - 設 global score 為 0
 - 設 Label1 文字 為 ""
 - 設 Button_START 文字 為 "再玩一次"

程式方塊區

工作面板：程式編輯區

Demo - 打地鼠遊戲

- 外觀編排與程式設計



The screenshot shows the Scratch IDE interface for a project named 'easy_mashmole'. The workspace contains several code blocks:

- 初始化全域變數 score 為 0** (Initialize global variable score to 0)
- 當 Button_START 被點選** (When Button_START is clicked):
 - 執行 **設 Clock1 啟用計時 為 true** (Set Clock1 timer to true)
 - 執行 **設 Clock2 啟用計時 為 true** (Set Clock2 timer to true)
- 當 ImageSprite1 被觸碰** (When ImageSprite1 is clicked):
 - 執行 **呼叫 ImageSprite1 移動到指定位置** (Call ImageSprite1 to move to specified position):
 - x座標: **整數亂數從 0 到 319** (Integer random number from 0 to 319)
 - y座標: **整數亂數從 0 到 319** (Integer random number from 0 to 319)
 - 執行 **呼叫 Sound1 播放** (Call Sound1 to play)
 - 執行 **設 global score 為 求 global score + 1** (Set global score to global score + 1)
 - 執行 **設 Label1 文字 為 求 global score** (Set Label1 text to global score)
- 初始化全域變數 clock1_time 為 10** (Initialize global variable clock1_time to 10)
- 當 Clock1 計時** (When Clock1 timer):
 - 執行 **呼叫 ImageSprite1 移動到指定位置** (Call ImageSprite1 to move to specified position):
 - x座標: **整數亂數從 0 到 319** (Integer random number from 0 to 319)
 - y座標: **整數亂數從 0 到 319** (Integer random number from 0 to 319)
 - 執行 **設 global clock1_time 為 求 global clock1_time - 1** (Set global clock1_time to global clock1_time - 1)
 - 執行 **設 Button_START 文字 為 求 global clock1_time** (Set Button_START text to global clock1_time)

修改打地鼠遊戲

- 開啟打地鼠遊戲檔：<https://goo.gl/VdW1S6>
- 增加一隻兔子，兔子也會隨機出現於螢幕，當點擊兔子時，總分扣3分。



參考資料

- 網站

- App Inventor官方網站 **英**

- <http://appinventor.mit.edu/explore/>

- App Inventor 中文學習網 **中**

- <http://www.appinventor.tw/>

- 離線版教學 **中**

- <https://sites.google.com/site/edreamer/app-inventor-2/app-inventor-2-server-setup>





Scratch延伸操作： S4A



Scratch延伸操作：S4A

- Scratch for Arduino

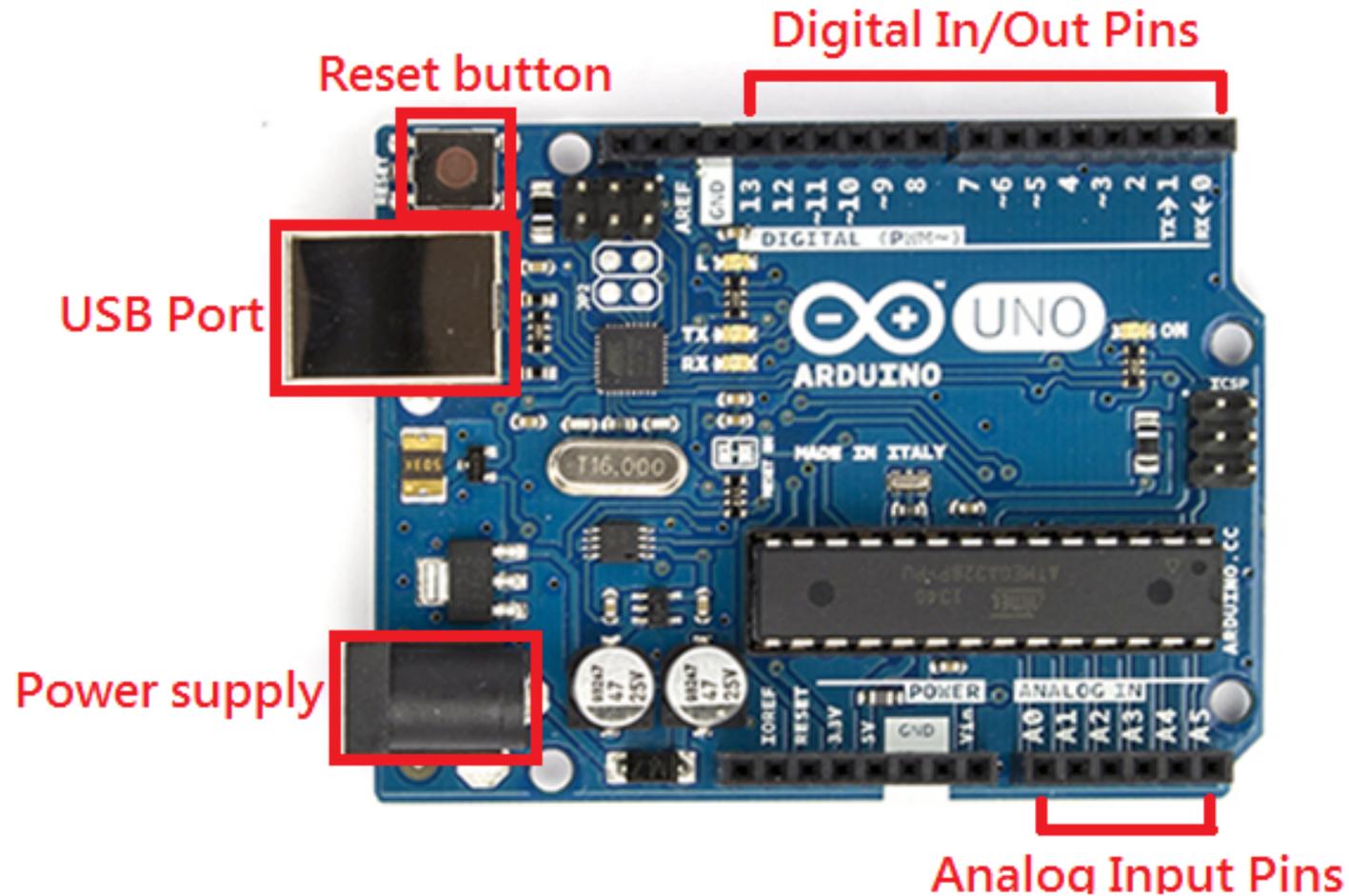
- 根據Scratch衍生的視覺化程式語言
- 歐盟Citilab團隊發展
- 具Scratch 的功能，且可與Arduino連接

- Arduino

- 開放原始碼的單晶片微控制器
- 建構於簡易輸出/輸入介面板



Arduino 介面



S4A設計介面

The screenshot displays the S4A 1.6 software interface, which is based on Scratch. The interface is divided into three main sections:

- 程式方塊區 (Block Area):** Located on the left, it contains various programming blocks for controlling the Arduino board, such as "value of sensor", "digital on/off", "analog value", "motor on/off", "motor direction", "motor angle", "reset actuators", "show board", "hide board", "board go to x: 0 y: 0", and movement blocks like "移動 10 步", "旋轉 15 度", and "旋轉 15 度".
- 程式編輯區 (Code Editor):** The central area where the code is written. It shows a script for an "Arduino1" board with coordinates (x: 3, y: -5) and a direction of 90 degrees. The code includes blocks for "digital 13 on", "digital 13 off", "analog 9 value 255", "motor 8 off", "motor 8 direction clockwise", "motor 8 angle 180", "reset actuators", "stop connection", "resume connection", "show board", "hide board", and "board go to x: 0 y: 0".
- 舞台編輯區 (Stage Area):** The right side of the interface, which displays the "Arduino Searching" window. This window shows the status of various pins: Analog0-5 (all 0), Digital2 (false), and Digital3 (false). Below this, there is a 3D model of an Arduino Duemilanove board. The stage area also includes a "新增角色" (Add Character) button and a "舞台" (Stage) label.

程式方塊區

程式編輯區

舞台編輯區

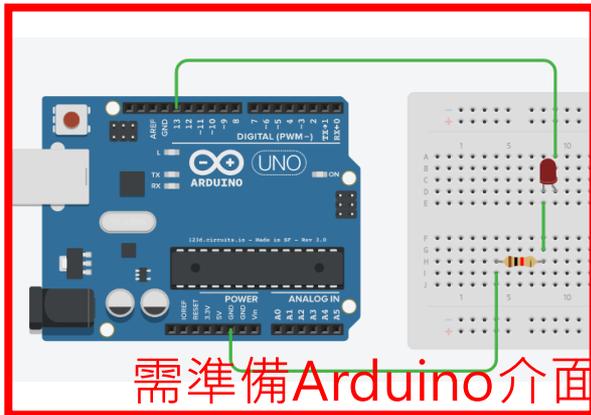
S4A設計介面

- 只有三種角色新增方法才可以與Arduino互動



眼明手快

- 遊戲規則：
 - 有紅色與綠色二顆LED，隨機亮燈
 - 綠燈亮起時，按下鍵盤的a鍵，獲得1分
 - 紅燈亮起時，按下鍵盤的a鍵，則扣5分



需準備Arduino介面控制板

針對Arduino介面控制板進行程式撰寫



