

Scratch3 AI 教育版

操作教學

[ML2Scratch](#) · [Facemesh2Scratch](#) · [Handpose2Scratch](#)

AI 擴充模組教學

適用版本：Scratch3-AI-Edu 1.4.7-beta（可攜版）

目錄

目錄	2
第一章 認識 Scratch3 AI 教育版	4
1.1 軟體說明	4
1.2 使用前的準備	4
1.3 啟動軟體	4
1.4 載入 AI 擴充模組	6
1.5 模組背後的原理	7
第二章 ML2Scratch：機器學習模組	9
2.1 功能介紹與運作原理	9
2.2 積木一覽	10
2.3 基礎範例：猜拳辨識器	13
2.4	152.5
17 第三章 Facemesh2Scratch：臉部網格偵測模組	34
3.1 功能介紹	34
3.2 積木一覽	35
3.3 常用臉部關鍵點對照	36
3.4 範例操作：小丑鼻貼紙	38
3.5 延伸範例 - 張嘴偵測	40
第四章 Handpose2Scratch：手部姿勢偵測模組	43
4.1 功能介紹	43
4.2 積木一覽	44
4.3 21 個手部關節點對照表	45
4.4 範例操作：食指空中畫筆	46
4.5 延伸範例	48

第一章 認識 Scratch3 AI 教育版

1.1 軟體說明

Scratch3 AI 教育版保留了原本 Scratch 的所有功能（角色、造型、聲音、積木程式等），並額外加入了三個 AI 人工智慧擴充模組，可以在沒有網路連線下，透過軟體在專案中增加「機器學習」、「臉部偵測」與「手部偵測」的功能：

- **ML2Scratch（機器學習）**：讓電腦「看圖學習」。拍下不同的影像並貼上標籤，讓電腦學會分辨它們，例如分辨剪刀、石頭、布。
- **Facemesh2Scratch（臉部網格偵測）**：透過攝影機即時偵測臉部的關鍵點的位置，此版本僅針對幾個臉部主要位置進行篩選，方便使用。
- **Handpose2Scratch（手部姿勢偵測）**：透過攝影機即時偵測一隻手的 21 個關節點位置，透過關節位置進行互動操作功能。

1.2 使用前的準備

- Windows 電腦，效能越好、整體操作會更流暢。
- 網路攝影機（筆電內建鏡頭或外接皆可）。三個 AI 模組都需要攝影機才能運作。
- 建議在光線充足、背景單純的環境下使用，辨識效果會較好。

1.3 啟動軟體

1. 找到 Scratch3-AI-Edu-1.4.7-beta-portable.exe 檔案，滑鼠左鍵快速點兩下執行。

· **建議**：把這個 .exe 複製到一個「路徑短、且非中文」的資料夾（例如 C:\Scratch3AI\）再執行，可避免少數電腦因路徑過長或權限問題啟動失敗。

2. 第一次啟動可能需要等待數秒到數十秒，會有等待提醒畫面出現，請耐心等待主畫面出現。



3. 若出現「Windows 已保護您的電腦」的 Windows SmartScreen 藍色警告視窗，請點「其他資訊」→「仍要執行」。

因為這個程式沒有購買數位簽章，並非病毒警告。這個畫面只會出現一次，只是「未簽章程式」的例行提醒。

4. 啟動後，可能會跳出這個視窗 Windows Defender 防火牆「安全性警示」

「Windows 安全性警示 / Windows Security Alert」

Windows Defender 防火牆已封鎖此應用程式的部分功能。

下方有兩個勾選框：☐ 私人網路 ☐ 公用網路。

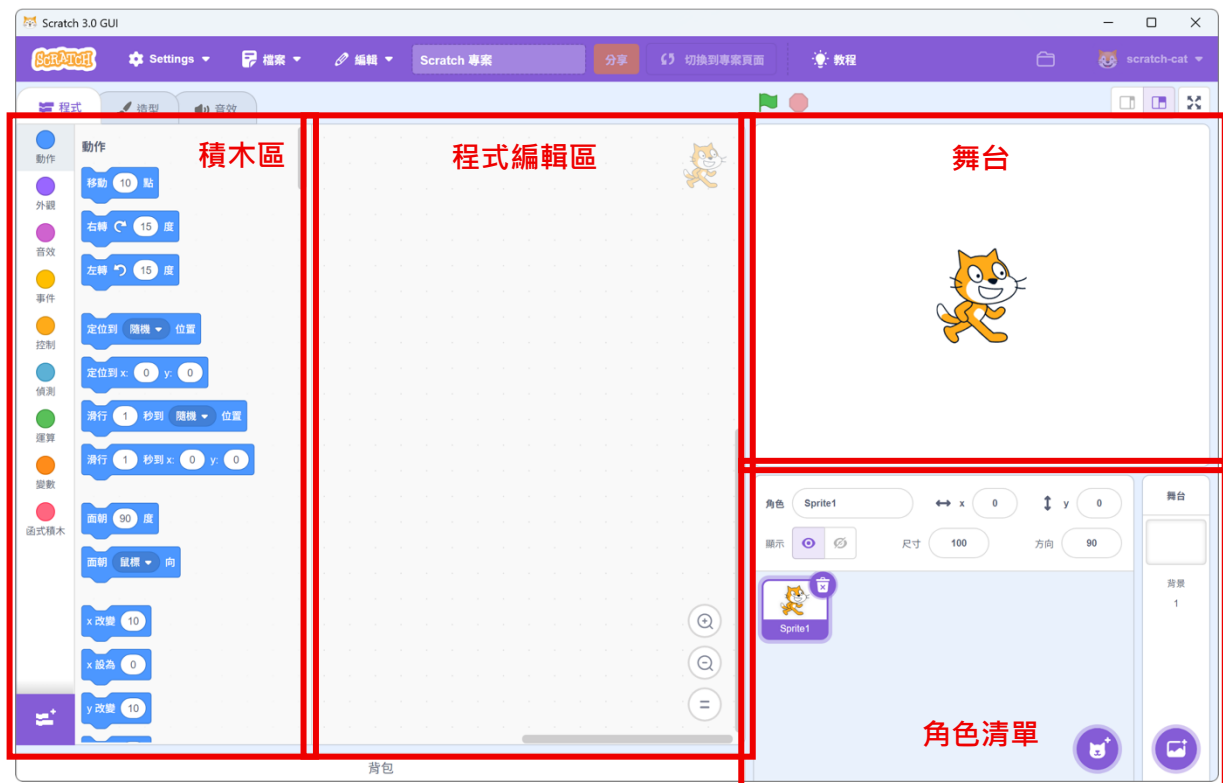
這個軟體是 完全離線 的，它 不會連到網際網路。但是為了讓 Scratch 編輯器正確運作，程式在啟動時會在你自己的電腦內部開一個小型網頁伺服器，網址固定是 `http://127.0.0.1:<隨機埠號>`。Windows 防火牆只要偵測到「有程式開了一個監聽用的網路連接埠」，不管它連不連外網，都會跳出這個警示 來問你要不要允許。所以這是正常現象，不是中毒，也不是程式有問題。

處理方式：勾選「私人網路」，按「允許存取」或直接按「取消」或關掉視窗。

例外狀況 — 如果按了允許／取消後，Scratch 卻是一片白畫面或一直轉圈：

通常是 第三方防毒軟體（如卡巴斯基、諾頓、ESET、360）或公司／學校的資安政策把本機伺服器也擋掉了。請到該防毒軟體的「防火牆 / 應用程式控制」中，把 Scratch3-AI-Edu-...portable.exe 加入 允許清單 / 例外，再重開程式。

5. 進入主畫面後，介面與一般的 Scratch 3 相同：左側是積木區、中間是程式編輯區、右上是舞台、右下是角色清單。



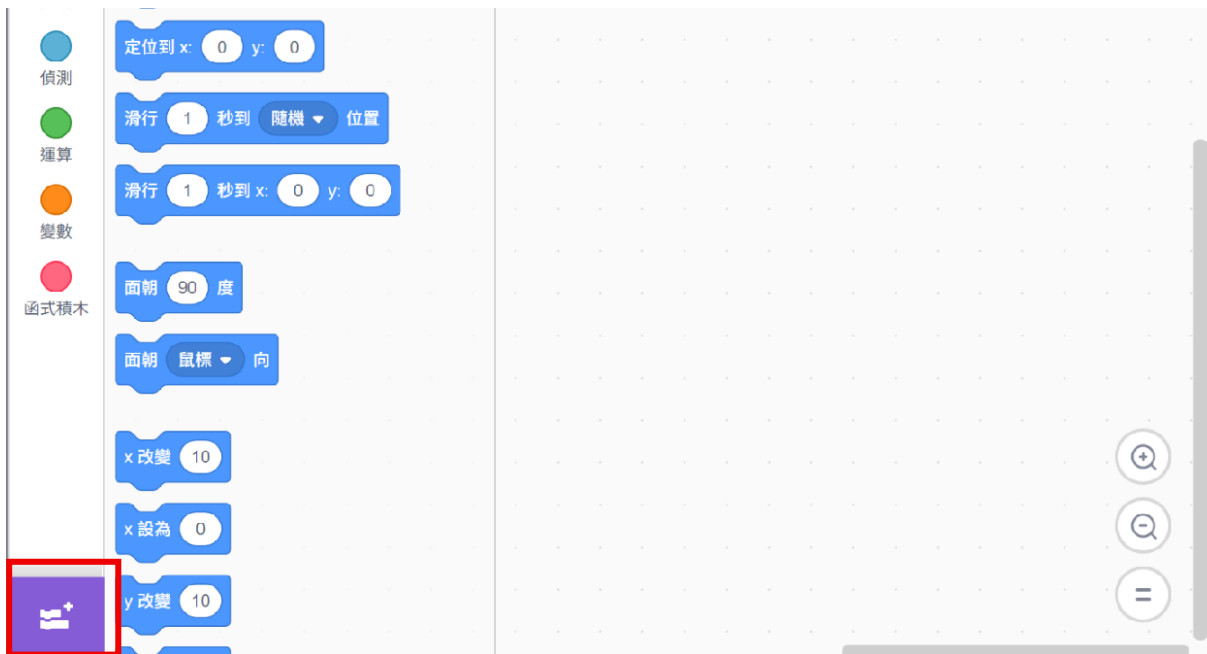
1.4 載入 AI 擴充模組

三個 AI 模組屬於「擴充功能」，需要手動加入後才會出現在積木區。步驟如下：

1. 點選畫面左下角的「添加擴展」按鈕。
2. 在擴充功能清單中找到 ML2Scratch、Facemesh2Scratch 或 Handpose2Scratch，點一下即可加入。
3. 第一次用到視訊時，Windows 或瀏覽器核心可能詢問是否允許使用相機，請選允許。另外請到 Windows 設定 → 隱私權與安全性 → 攝影機，確認「讓桌面應用程式存取您的攝影機」為開啟。

□ 一次只開一個 Scratch3-AI-Edu 視窗。本軟體設計為「單一執行個體」，若重複開啟，兩個視窗會搶同一支相機，導致畫面卡住。關閉舊視窗再開新的即可。另外，外接 USB 攝影機時，在多次操作、切換鏡頭後，有可能會發生無法正常運作的情況（這是原生模組的問題），請重新插拔 USB 就可以解決。

4. 積木區左側會多出該模組的分類圖示，點選即可看到所有新積木。





□ 第一次載入會「卡住」是正常的！Facemesh2Scratch 與 Handpose2Scratch 載入時會跳出提示視窗：「Setup takes a while. The browser will get stuck, but please wait.」（準備需要一些時間，畫面會暫時卡住，請稍候。）這是因為軟體正在載入 AI 模型，依電腦效能而有所不同。請按下確定後耐心等待，不要重複點擊或關閉軟體。

1.5 模組背後的原理

在開始操作前，先認識幾個關鍵概念，之後的操作會更有感覺：

- 機器學習（ML2Scratch 使用）：電腦不是被「寫死」規則，而是從提供的範例中自己歸納規則。您給它 20 張石頭的照片，它就學會「石頭長這樣」。範例越多、越多樣，學得越好。這種「給範例、貼標籤」的方式稱為監督式學習。
- 關鍵點偵測（Facemesh／Handpose 使用）：AI 模型事先已被訓練好，能在畫面中找出臉或手的特定位置（稱為關鍵點或地標），並回報每個點的座標。不需要訓練它，載入後立刻可用。

簡單來說：ML2Scratch 是「需要教它」的模組，Facemesh2Scratch 和 Handpose2Scratch 是「已經學會、直接使用」的模組。

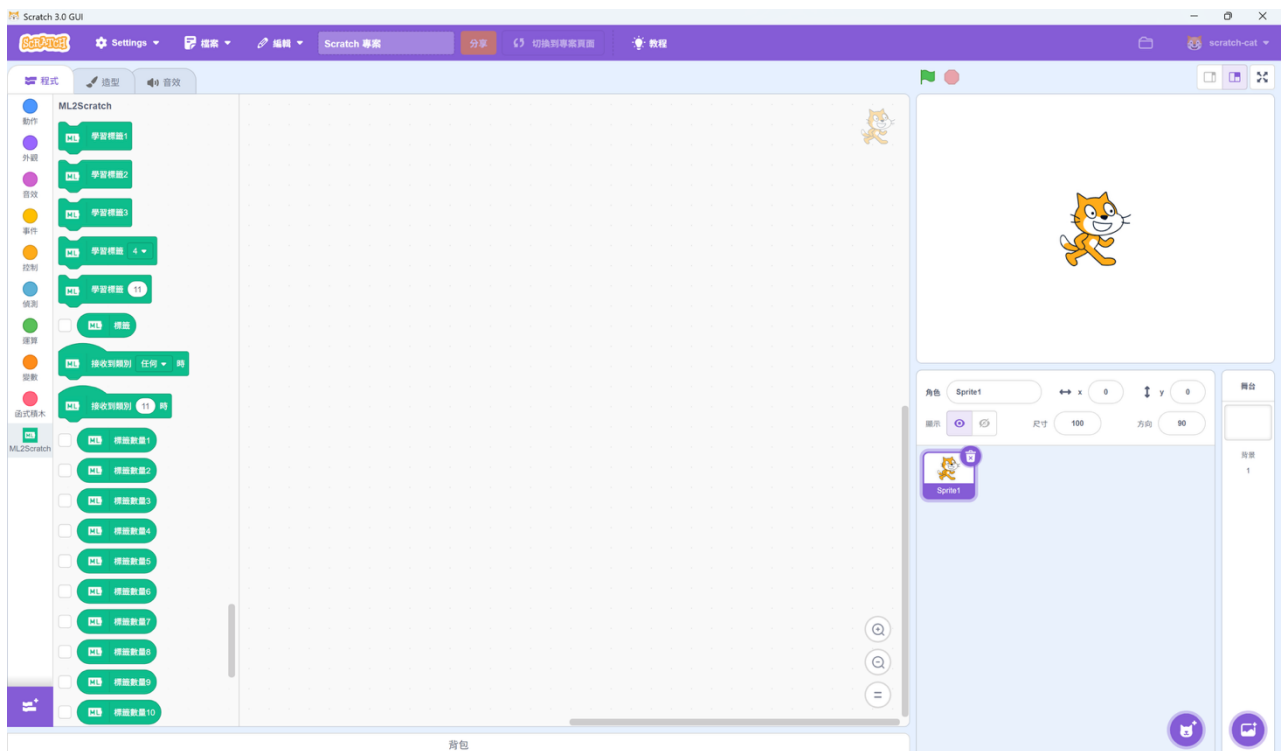
第二章 ML2Scratch：機器學習模組

2.1 功能介紹與運作原理

ML2Scratch 讓您在 Scratch 中完成機器學習流程：收集資料 → 訓練模型 → 使用模型辨識。它的用途是「影像分類」。把攝影機看到的畫面分成您定義的類別（稱為「標籤」，以數字 1、2、3…命名）。

運作原理：每按一次「學習標籤」積木，軟體就把攝影機當下的畫面拍下來、交給模型抽取影像特徵，存進對應標籤的影像庫。之後進行辨識時，軟體會把新畫面與所有影像比較，找出最相似的標籤作為辨識結果。整個過程都在本地端完成。

能做什麼？例如：辨識剪刀石頭布做猜拳遊戲、辨識手寫數字、辨識不同物品做垃圾分類教學、辨識姿勢或表情來控制遊戲角色等。



電腦是怎麼分辨的？

每按一次「學習標籤」，電腦就把當下的畫面轉成一串數字（影像的「特徵」）存進該標籤的影像庫；辨識時，再把新畫面跟所有存過的影像比一比，「最像誰，就判給誰」。

- 為什麼影像要拍多張、還要變換角度：存的影像越多、變化越豐富，電腦比對時越容易找到相似的，辨識就越準。
- 為什麼換人、換背景會變不準：電腦比的是「整個畫面」，背景和人也算在內；環境變了就補拍幾張新影像即可。

- 為什麼要多訓練一個「沒有目標」的標籤：電腦永遠會挑一個「最像的」回答，不會說「我不知道」；給它一個背景標籤，沒東西時它才有正確答案可選。
- 重要動作前建議搭配「穩定判斷機制」：因為電腦永遠有答案，過渡瞬間可能閃過錯誤結果，同一結果持續 2-3 秒再行動較佳。

2.2 積木一覽

ML2Scratch 的積木依用途分為五大類。積木文字以軟體內建的繁體中文翻譯為準：

1. 訓練用積木：教電腦認識影像

積木	類型	功能說明
	指令型積木	把攝影機當下的畫面拍一張，存為「學習標籤 X」的學習影像。每點一次存一張，建議每個標籤收集約 30 張。
	指令型積木	同上，但透過下拉選單選擇標籤 4~10，可建立更多類別。
	指令型積木	同上，標籤編號可自由填入任何數字，不受選單限制。
	指令型積木	刪除學習影像。可選單一標籤（只清除該類）或「所有量」（全部歸零重來）。執行前會跳出確認視窗。
	指令型積木	清除任意編號標籤的影像。

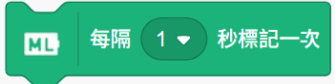
2. 辨識結果積木：讀取電腦的判斷

積木	類型	功能說明
	功能型積木	顯示每個標籤目前已訓練的影像張數。可勾選積木左側核取方塊，直接顯示在舞台上觀察。
	功能型積木	任意編號標籤的影像張數。
	事件型積木	當辨識結果「變成」指定標籤的那一刻，執行底下的程式。選「任何」則任何標籤改變都會觸發。這是撰寫互動程式最常用的積木。標籤編號可自由填入。

3. 學習資料的儲存與載入

積木	類型	功能說明
	指令型積木	把目前訓練好的所有影像匯成一個 .json 檔存到電腦。注意：學習資料不會存在 .sb3 專案檔裡，必須用此積木另外儲存！
	指令型積木	開啟上傳視窗，載入之前下載的 .json 檔，立刻恢復訓練成果，不必重新拍照訓練。

4. 辨識行為設定

積木	類型	功能說明
	指令型積木	關閉或開啟自動辨識。訓練期間先關閉分類可讓畫面較順暢。
	指令型積木	用途是設定模型執行「分類（推論）」的頻率。也就是每隔多少秒，把攝影機當下的畫面丟進已訓練好的模型判斷一次，並更新「標籤」回報積木的值、觸發「接收到類別」子積木。可選 1、0.5、0.2、0.1 秒。頻率越高反應越快，但電腦負擔越重。
 	指令型積木	選擇辨識輸入來源，主要分兩種： 1. 網路攝影機：直接使用鏡頭畫面 2. 舞台：辨識整個舞台的畫面（含角色與畫筆內容），適合手寫辨識等應用。
	指令型積木	電腦接了多顆攝影機時，切換要使用哪一顆。

5. 畫面顯示設定

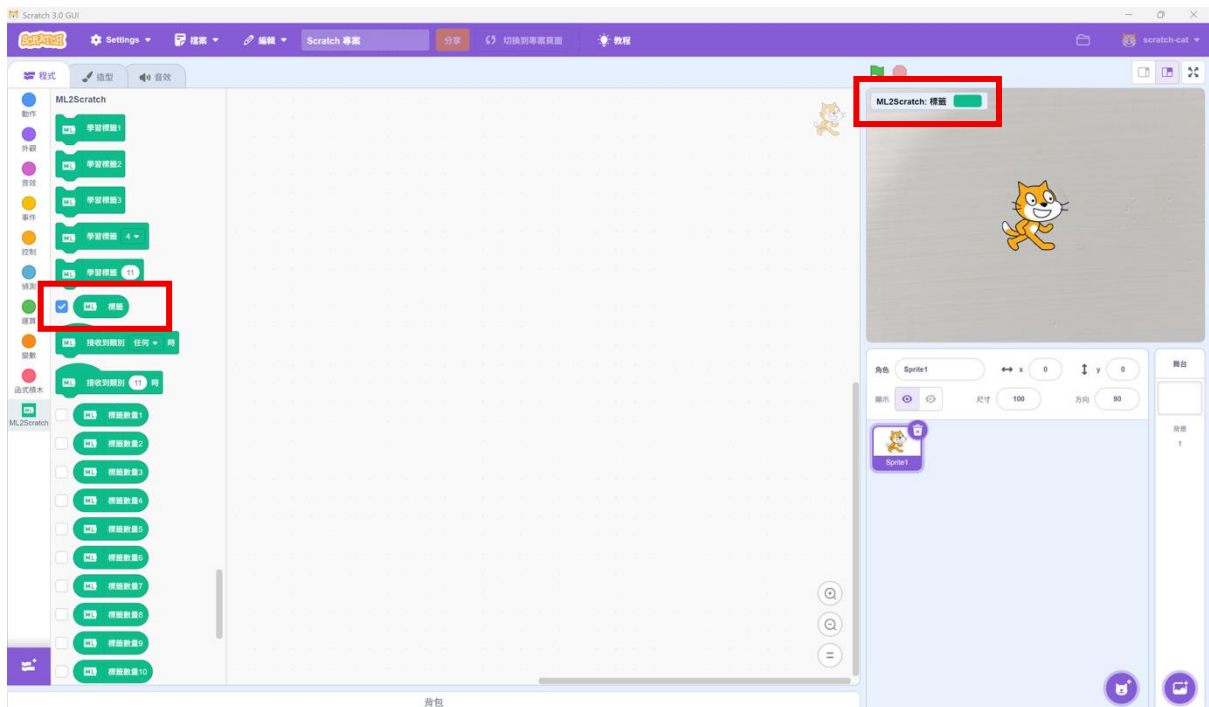
積木	類型	功能說明
	指令型積木	控制舞台上的攝影機畫面：開啟／關閉／翻轉（左右鏡像，如照鏡子效果）。
	指令型積木	調整攝影機畫面透明度（0＝完全清楚，100＝完全看不到）。讓角色更明顯時可調高。

2.3 基礎範例：猜拳辨識器

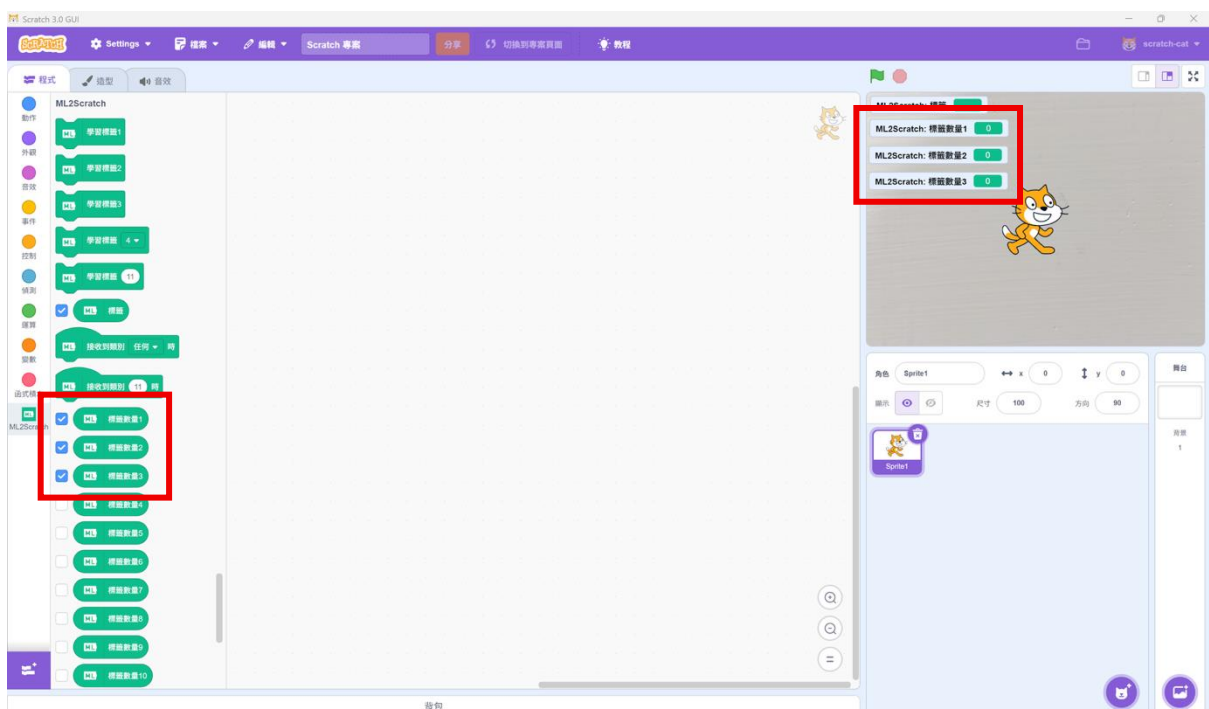
目標：訓練電腦分辨「石頭、剪刀、布」，出拳時讓角色說出答案。

步驟 1：準備

1. 載入 ML2Scratch 擴充模組並允許攝影機。
2. 在積木區點選 ML2Scratch 分類，勾選「標籤」積木左側的核取方塊，讓辨識結果顯示在舞台上。



3. 勾選「標籤數量 1」「標籤數量 2」「標籤數量 3」，方便觀察拍了幾張。

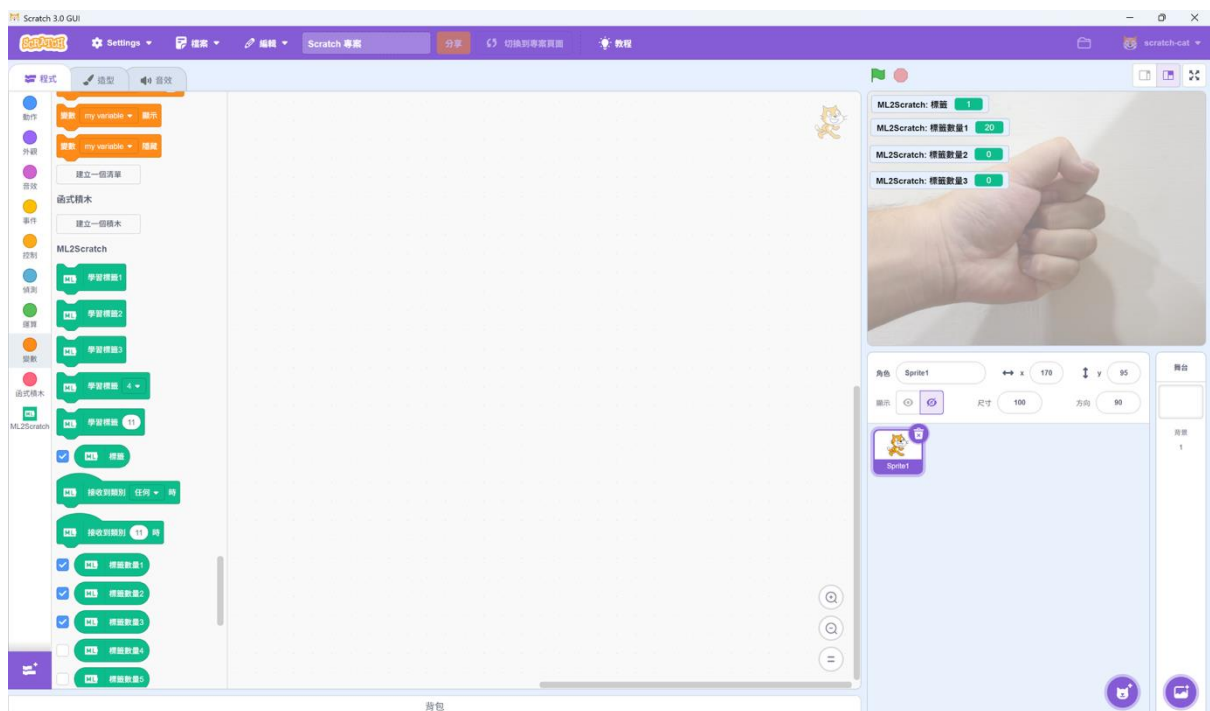
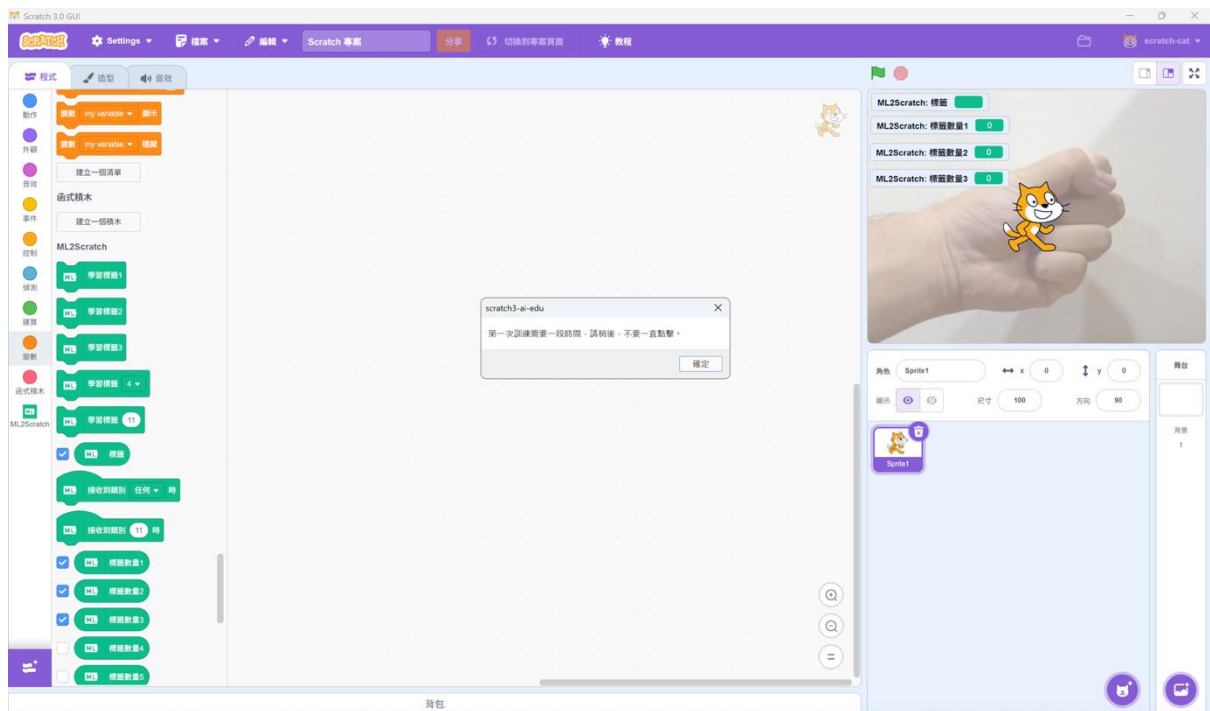


步驟 2：收集訓練資料

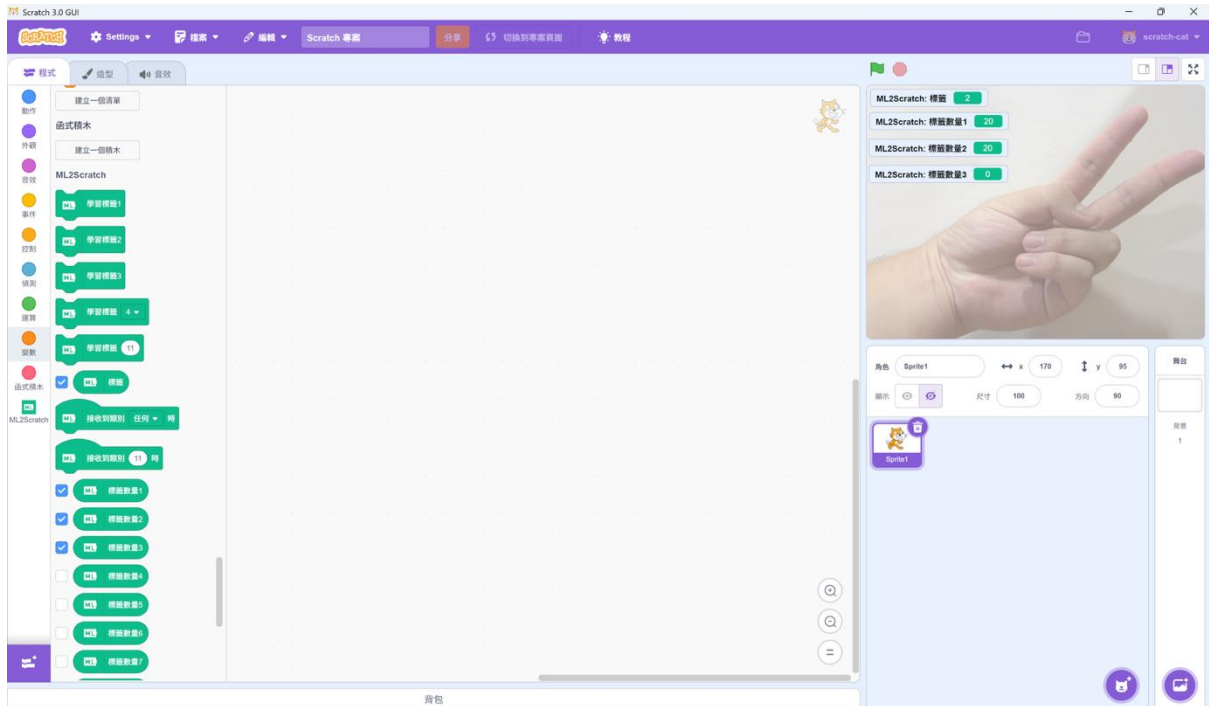
1. 對著鏡頭比出「石頭」，用滑鼠連續點擊「學習標籤 1」積木約 20 次。每點一次就拍下一張影像，「標籤數量 1」會跟著增加。點擊時稍微移動、旋轉手的位置與角度，影像越多樣，辨識越準。

□ 第一次點「學習標籤」會跳出提示

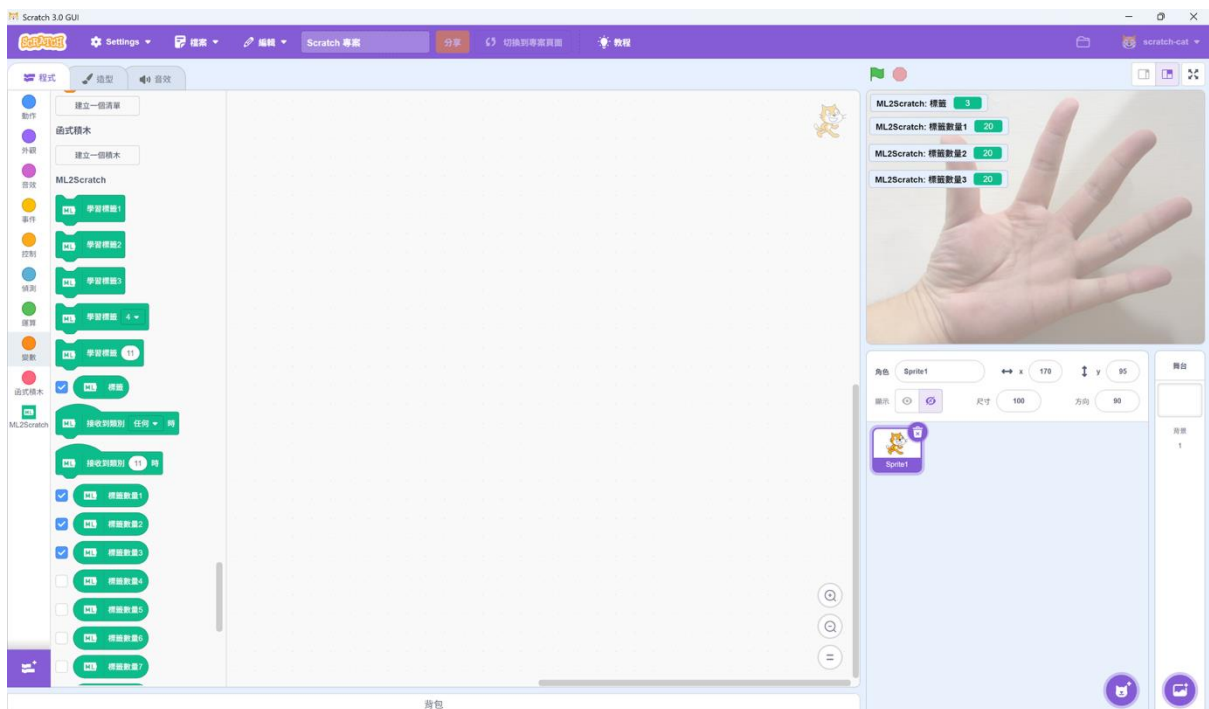
訊息內容：「第一次訓練需要一段時間，請稍後，不要一直點擊。」這是模型初始化，等它完成後再繼續點擊即可。



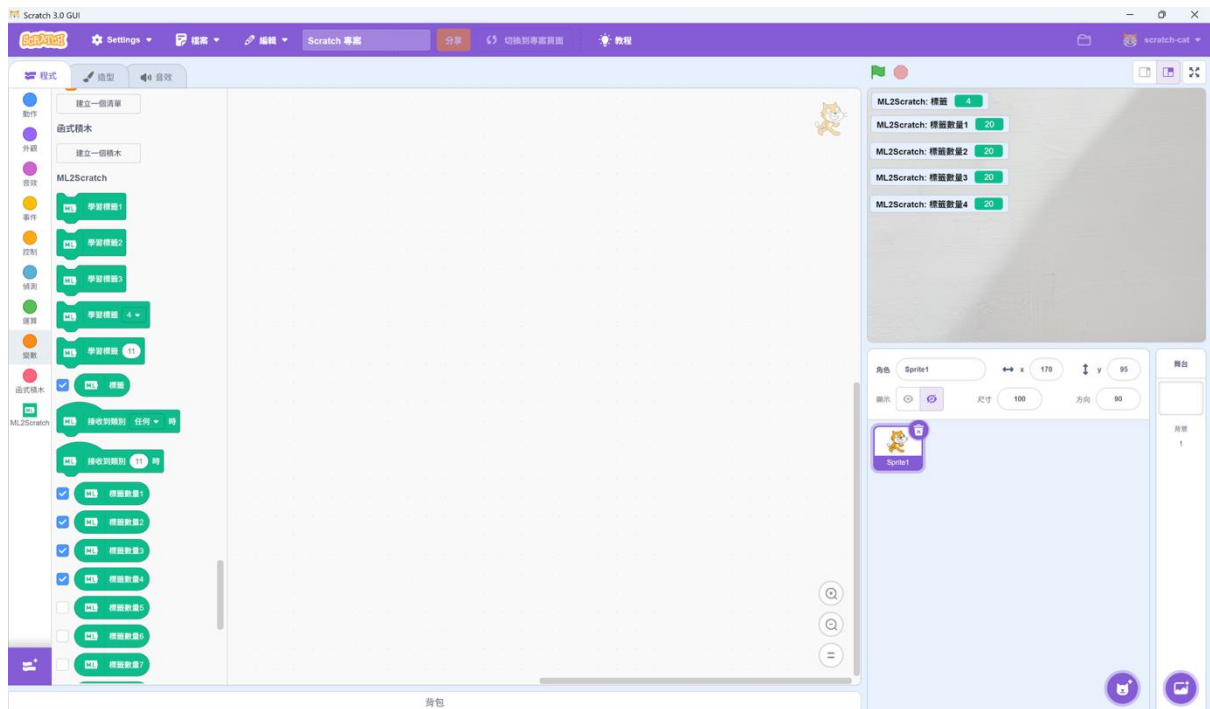
2. 比出「剪刀」，同樣點擊「學習標籤2」約20次。



3. 比出「布」，點擊「學習標籤 3」約 20 次。

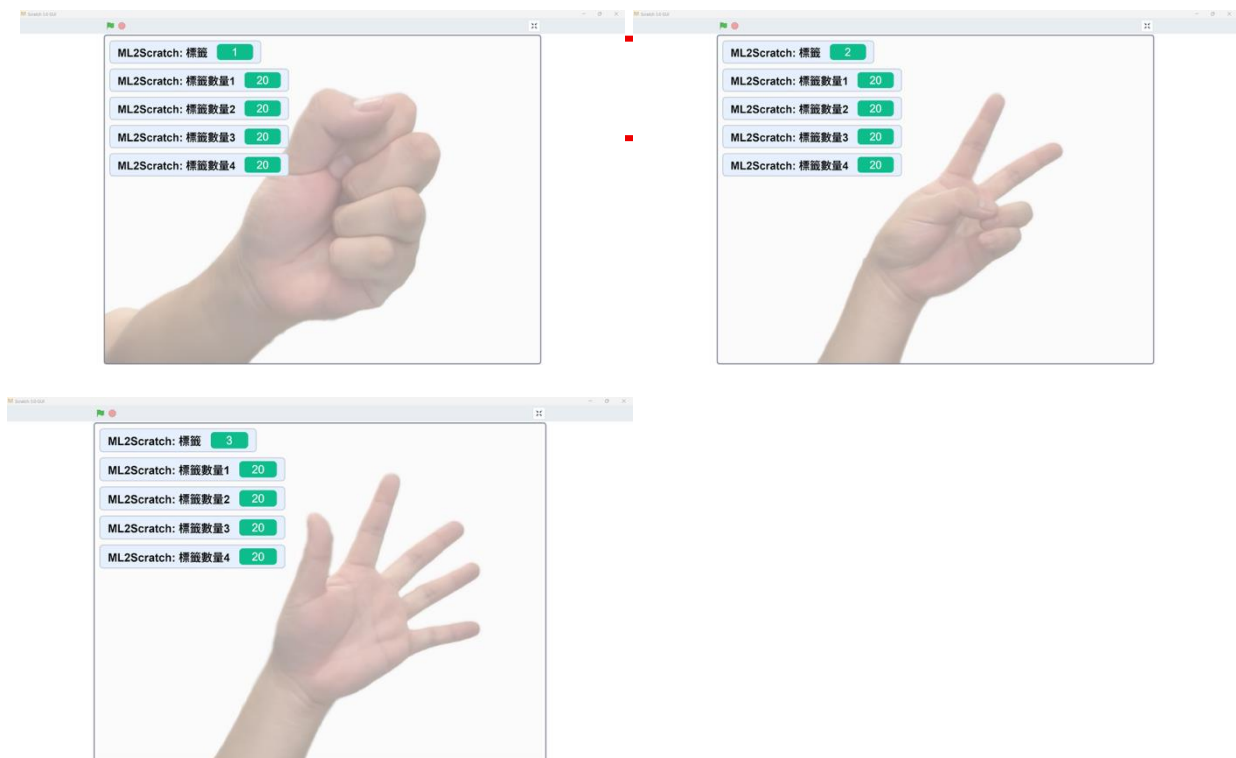


4. 把手移開畫面，點擊  約 20 次，教電腦認識「沒有出拳」的狀態，以降低未出拳時誤判的狀況。



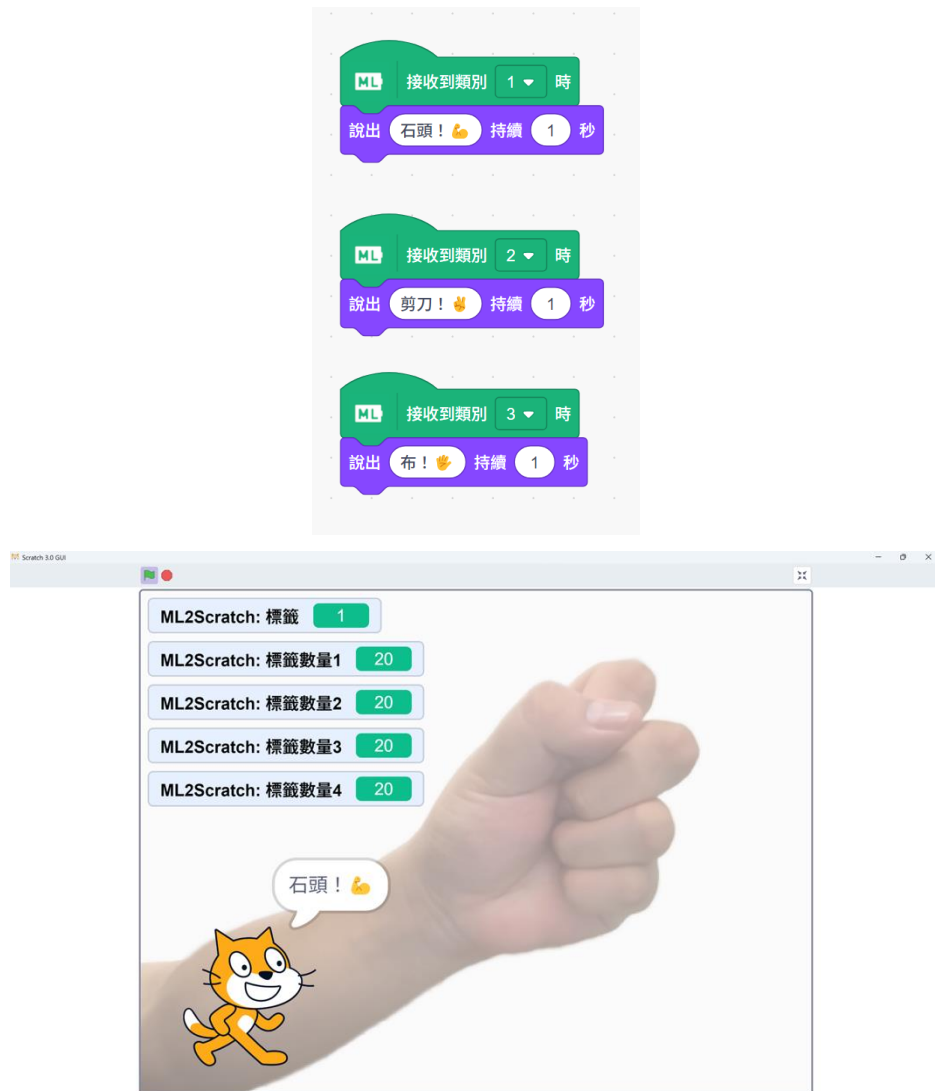
步驟 3：測試辨識效果

訓練完成後，對鏡頭輪流比出三種手勢，觀察舞台上「標籤」的數值：出石頭時應顯示 1、剪刀顯示 2、布顯示 3。若常常判斷錯誤，請針對容易誤判的手勢補拍更多影像，或按「重置」重新訓練。



步驟 4：加上互動程式

使用「接收到類別」帽子積木，讓角色即時反應。範例程式如下：



► 積木程式詳細說明

- 這是三段各自獨立的程式，都以  這類型積木開頭——該積木是程式的「開關」，平時待命，事件發生的那一刻才執行下面的積木。

-  的觸發時機是辨識結果「變成」1 的那一瞬間：只觸發一次，維持同一手勢期間不會重複觸發；要把手放下再比一次，才會再觸發。

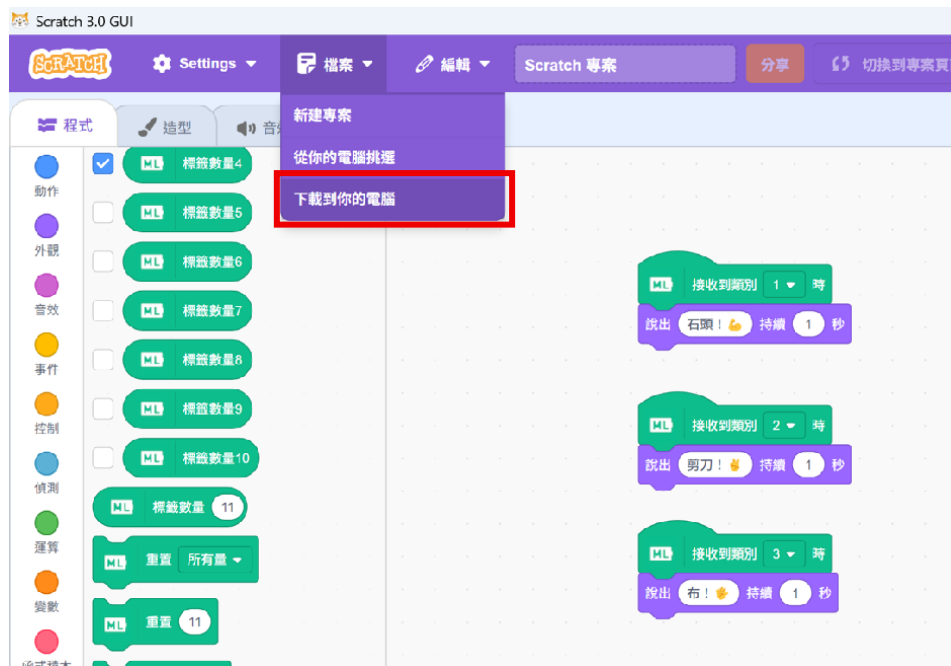
- 三段程式分別對應標籤 1（石頭）、2（剪刀）、3（布），觸發後用

 讓角色即時報出辨識結果。

2.4 儲存專案與學習資料

ML2Scratch 的專案需要儲存「兩個檔案」，缺一不可：

- 程式：功能表「檔案」→「儲存到電腦」，會存成 .sb3 檔。



- 學習資料：點擊「下載學習資料」積木，會存成 .json 檔。



ML_example.json

下次開啟 .sb3 專案時，點擊「上傳學習資料」積木載入 .json 檔，即可完整恢復訓練資料集，不必重新訓練。



選擇一個檔案，然後點擊上傳按鈕

選擇檔案 ML_example.json

上傳

關閉

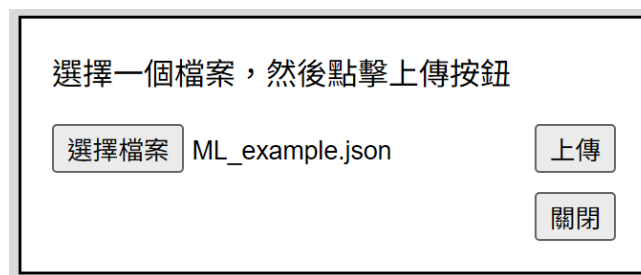
2.5 延伸範例

延伸範例 1：人機猜拳對戰（含勝負判斷與計分）

在基礎範例之上，讓電腦隨機出拳、自動判斷輸贏並計分。這個挑戰同時練習「變數」與「條件判斷」。

步驟 1：準備

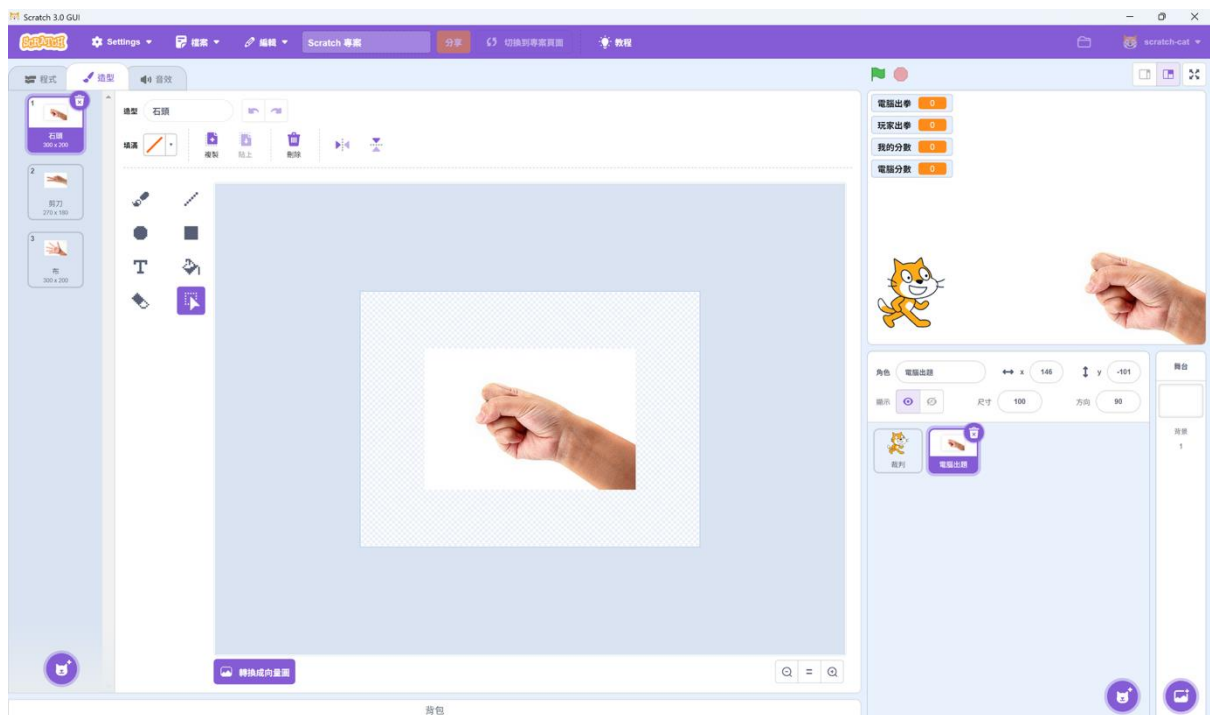
1. 匯入先前基礎範例的訓練資料



2. 在「變數」分類點「建立一個變數」，依序建立四個變數：電腦出拳、玩家出拳、我的分數、電腦分數。



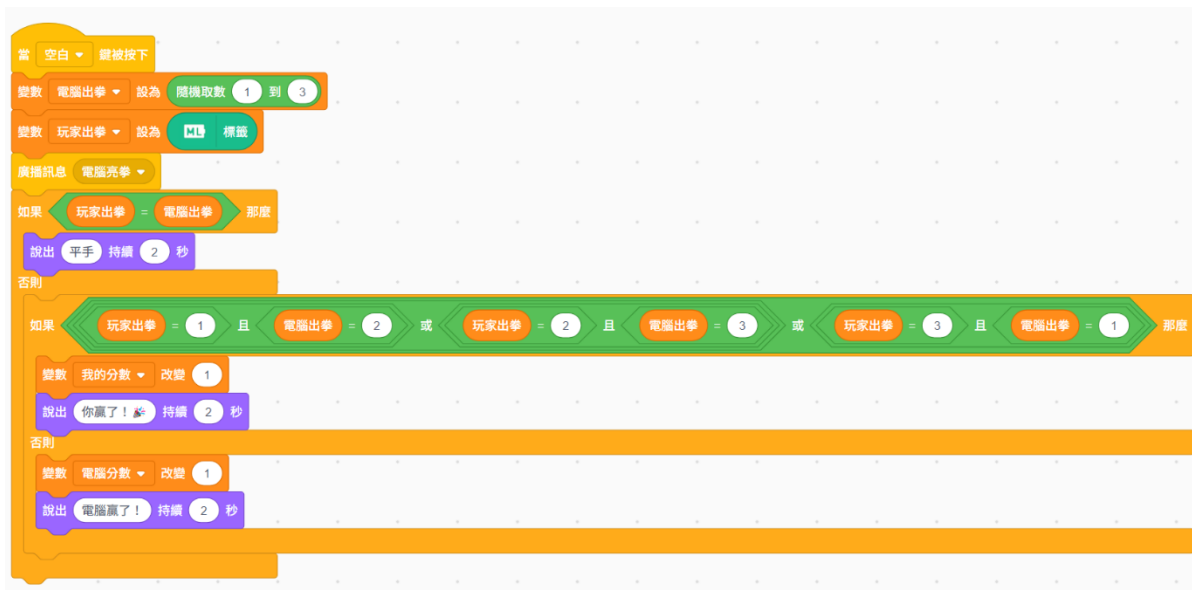
- 幫「電腦」新增一個角色，並為它準備三個造型：造型 1 畫石頭、造型 2 畫剪刀、造型 3 畫布。



- 勝負規則對照：1（石頭）勝 2（剪刀）；2（剪刀）勝 3（布）；3（布）勝 1（石頭），玩法為玩家先比好手勢、確認舞台上「標籤」顯示正確後，按下空白鍵定勝負。

步驟 2：判斷流程

- 裁判的程式：



► 積木程式詳細說明

- 觸發：整段程式由  啟動——玩家先比好手勢、確認舞台上「標籤」顯示正確後，按空白鍵定勝負。
- ：用「運算」類的隨機積木讓電腦公平出拳，1~3 分別代表石頭、剪刀、布。
- ：把 ML2Scratch 的「標籤」橢圓積木塞進「設為」的空格，等於在按鍵的瞬間「讀取玩家的出拳」——玩家本人不需要任何程式，攝影機＋訓練好的模型就是玩家的輸入裝置。
- ：通知電腦角色換造型亮拳。
- 第一層  判斷平手；否則進入第二層，用兩層「或」積木把三組獲勝組合  串在一起——「或」一次只能接兩個條件，所以三組條件需要兩層嵌套。成立則 ，不成立則 ，並各自說出結果。

2. 電腦亮拳程式，收到廣播後換成對應造型：



► 積木程式詳細說明

- ：這段程式平時不動作，收到裁判的廣播才執行。
- ：把變數「電腦出拳」塞進「換成造型」的空格，數值 1~3 會直接對應造型清單的第 1~3 個——所以造型順序必須排好：1 石頭、2 剪刀、3 布，順序錯了亮出來的拳就會錯。

3. 完成後即可進行測試

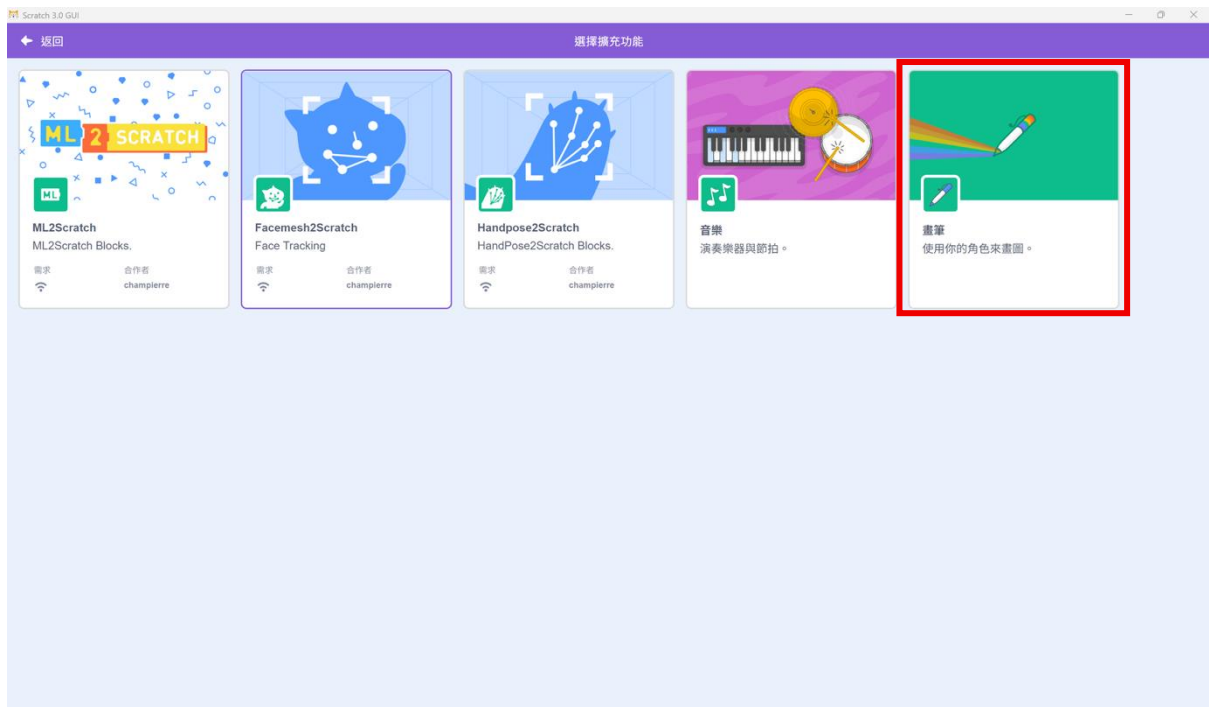


延伸範例 2：手寫數字辨識板

這個案例改用「舞台」作為辨識來源：學生用滑鼠在舞台上寫數字，讓電腦學會分辨「1」和「2」。它展示了 ML2Scratch 不只能辨識鏡頭畫面，任何舞台上的圖像都能學。

步驟 1：準備

1. 點「添加擴展」，加入內建的「畫筆」擴充模組。



2. 把舞台背景維持全白。

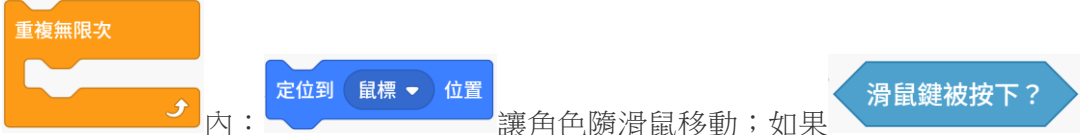
步驟 2：訓練資料與驗證

1. 寫字板程式（放在任一角色上）：



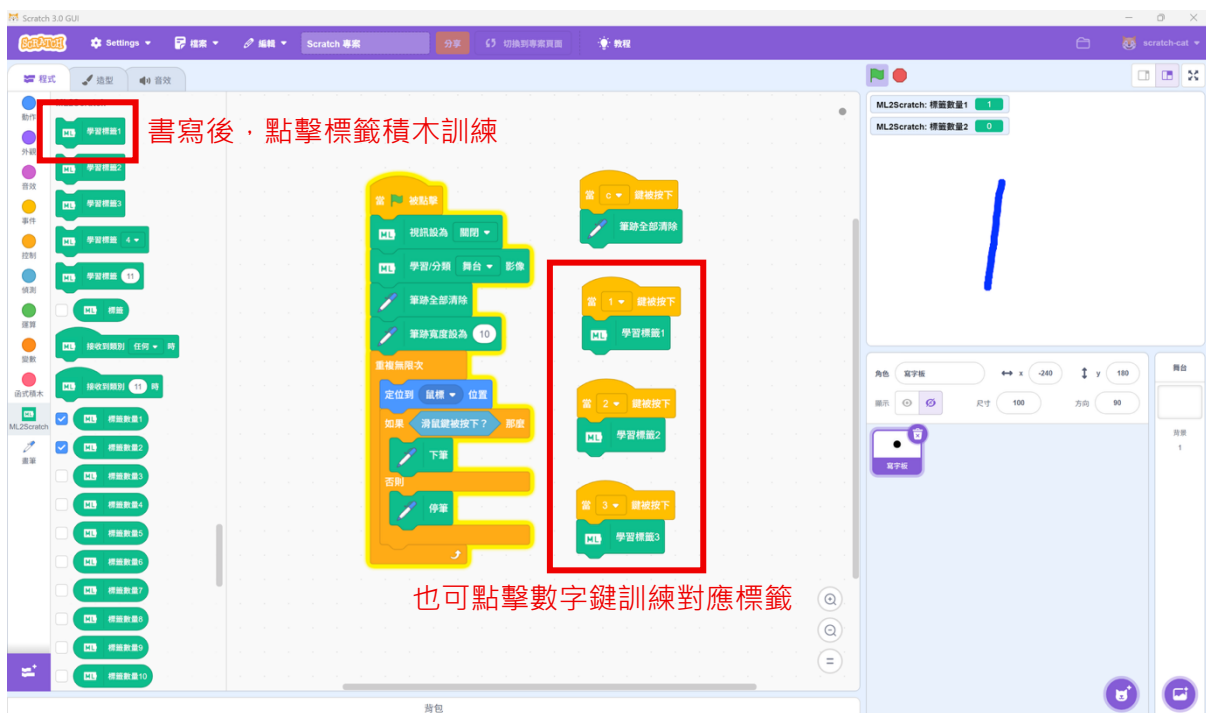
► 積木程式詳細說明

- ：關掉鏡頭畫面，舞台變成乾淨的白色畫布。
- ：把辨識來源從攝影機切換成「舞台截圖」，之後訓練與辨識抓的都是舞台上的筆跡。
- ：線條粗一點，在畫面中佔比較大，特徵較明顯，辨識效果較好。

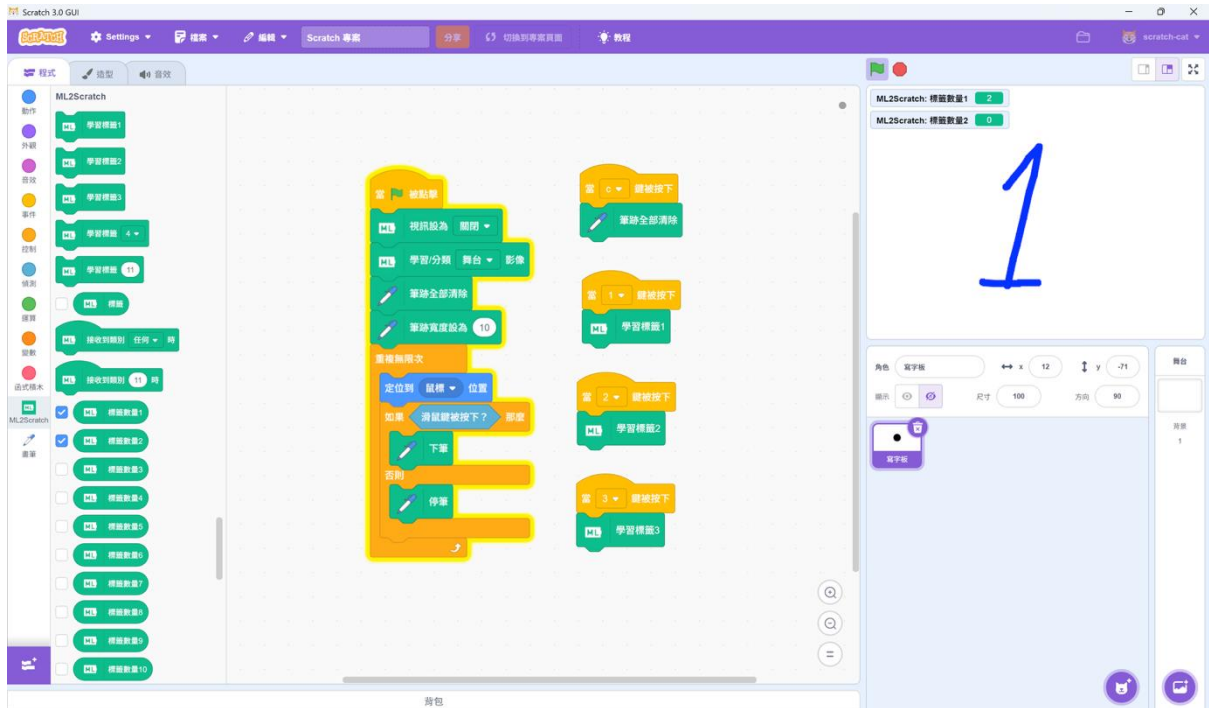
- 
 內：讓角色隨滑鼠移動；如果滑鼠鍵被按下？，則下筆，否則就停筆—按住左鍵拖曳就畫線、放開就停，滑鼠變成一支畫筆。

- 
 ：按 c 鍵清空畫布，準備寫下一個字。

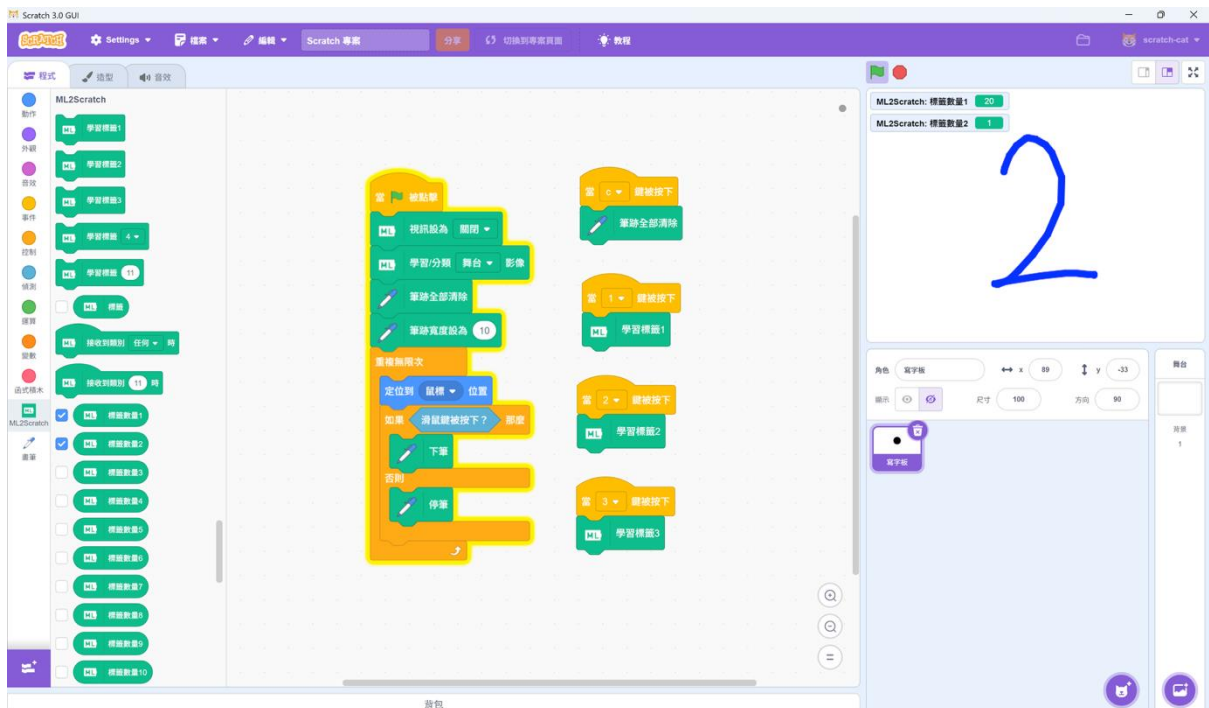
- 按綠旗啟動寫字板。在舞台上用滑鼠寫一個「1」，然後點擊「學習標籤 1」積木一次。亦可以設定點擊數字鍵時，啟動一次訓練。



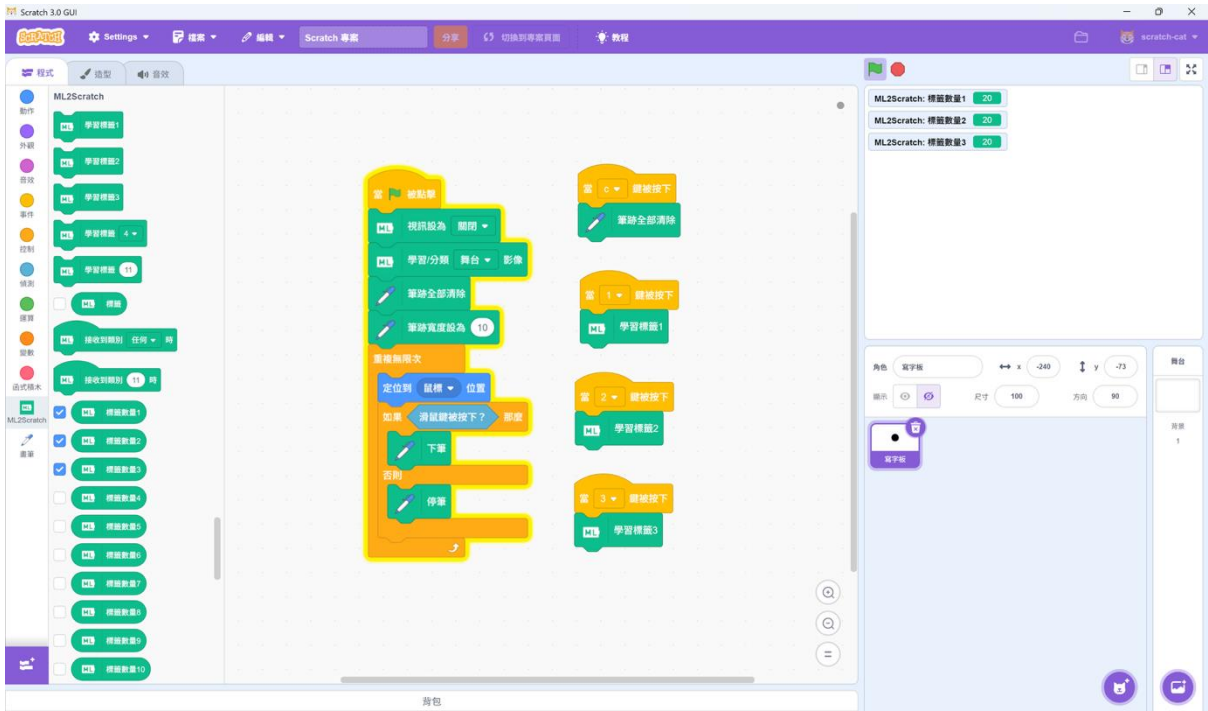
- 按 c 鍵清除，再寫一個長得不太一樣的「1」（位置、大小、斜度變化），再點「學習標籤 1」。重複收集約 20 個不同的「1」。



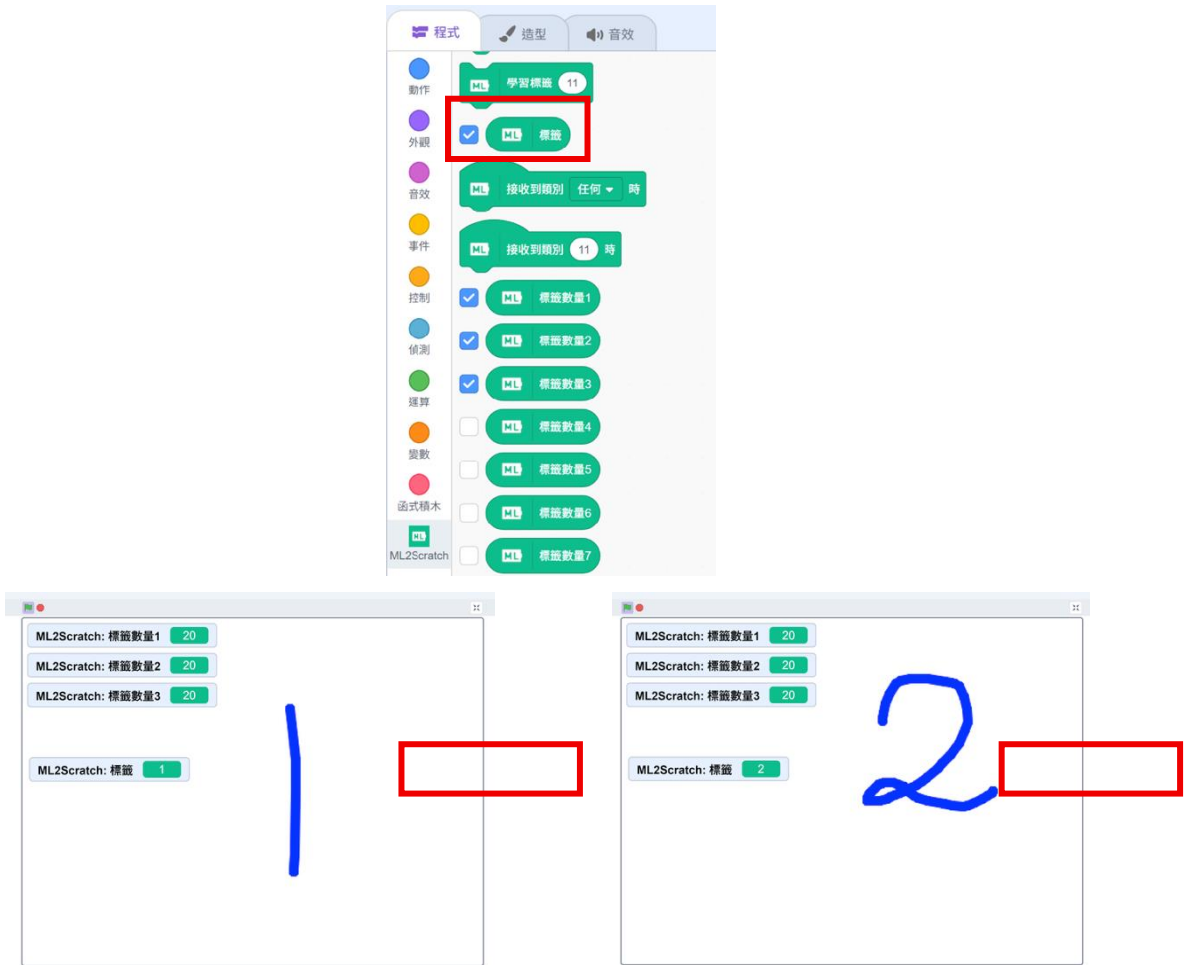
4. 用同樣方式寫 20 個不同的「2」，每寫一個點擊一次「學習標籤 2」。



5. 空白畫布也點「學習標籤 3」約 20 次，代表「還沒寫字」。



6. 寫一個新的數字，觀察「標籤」監視器是否正確判斷。



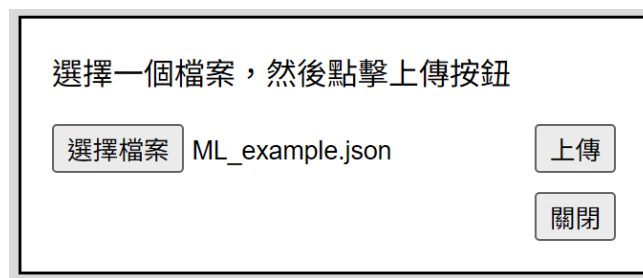
延伸範例 3：穩定判斷機制—同一結果持續 2 秒才算數

由於 ML2Scratch 的積木沒有提供「信心值（confidence）」，不管有多不確定，「標籤」都會回報一個最接近的答案。因此在手勢切換的過程中、或影像模糊的瞬間，辨識結果常會短暫閃過錯誤的標籤，如果程式一看到標籤就立刻反應，就會頻繁誤觸發。

為了解決此問題，我們將在程式中做出「持續時間確認」機制—同一個標籤必須連續維持一段時間（例如 2 秒）不變，才視為「確認的結果」並觸發動作；中途只要標籤一變，就歸零重新計時。

操作流程

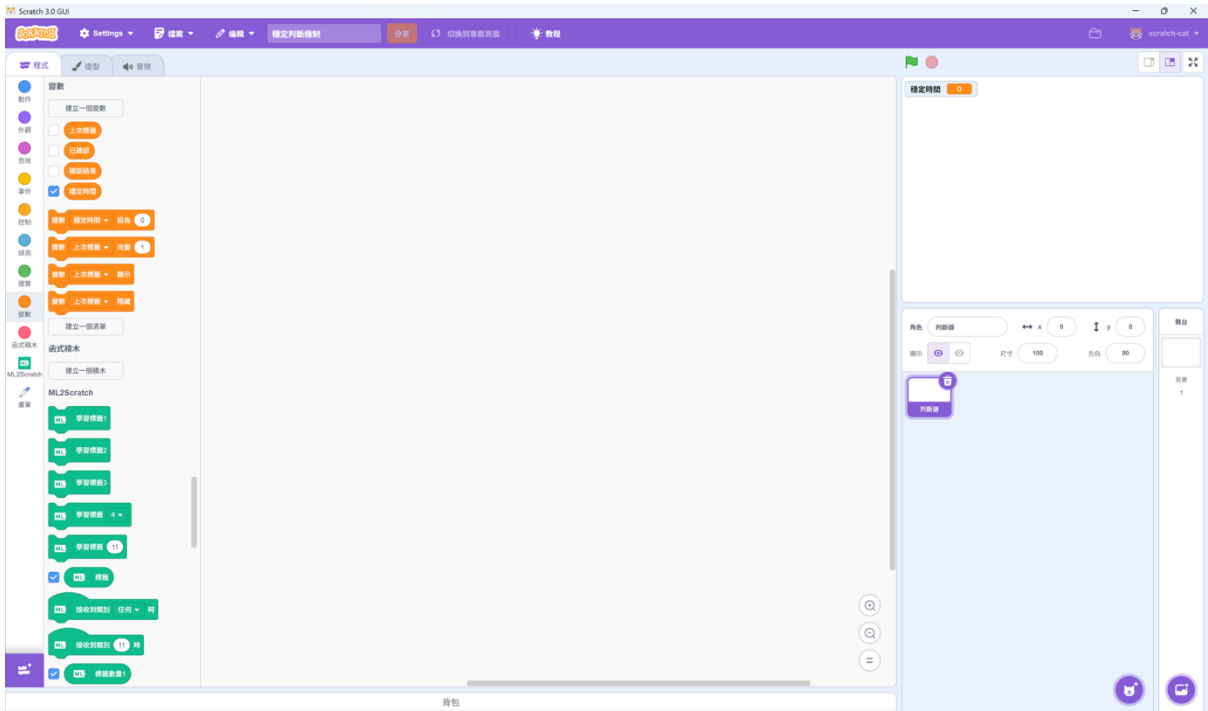
1. 匯入先前的訓練資料



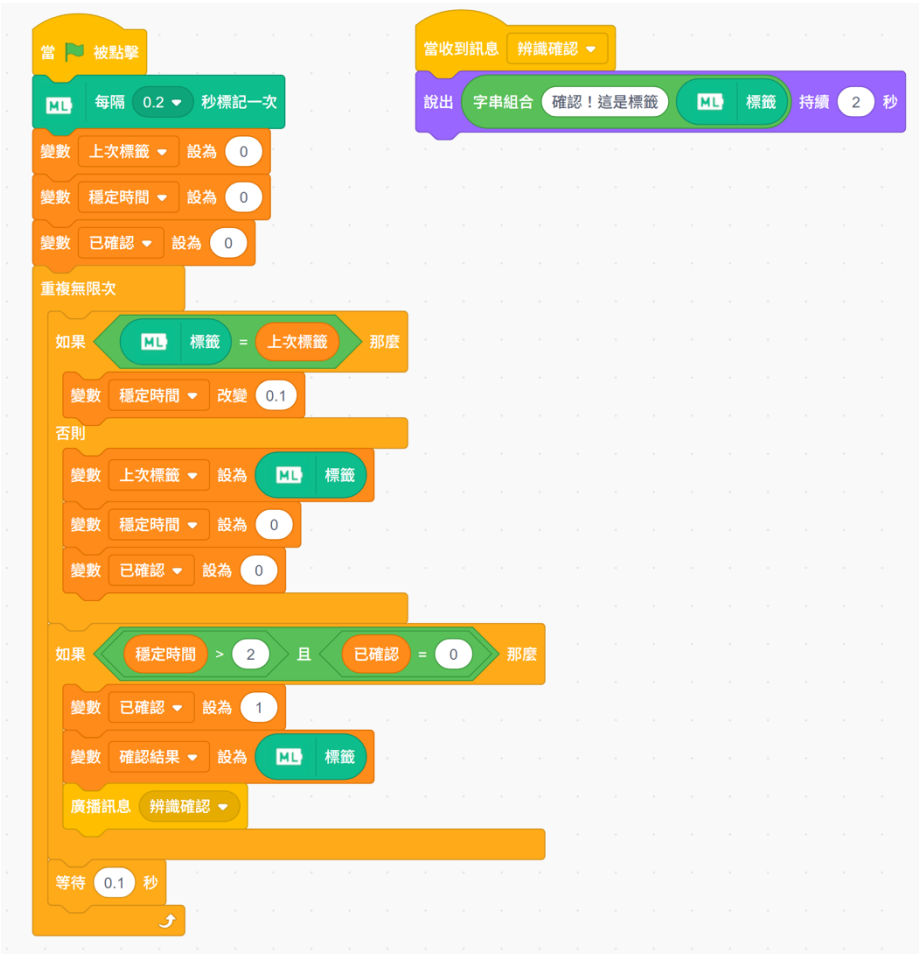
2. 建立四個變數：上次標籤、穩定時間、已確認、確認結果。勾選「穩定時間」顯示在舞台上，能看到計時進度。



3. 新增任一角色



4. 於專案中的角色上組合以下兩段程式：



► 積木程式詳細說明

- 初始化：綠旗按下後先把「上次標籤」「穩定時間」「已確認」歸零；



把辨識頻率調快，計時才靈敏。

- 主迴圈每 0.1 秒比對一次：現在的「標籤」和「上次標籤」一樣嗎？一樣則



；若不一樣則記下新標籤、「穩定時間」歸零、「已確認」歸零（重新開始計時）。

- 如果 ：同一結果連續撐過 2 秒才成立—過渡瞬

間閃過的錯誤標籤因為撐不過 2 秒，全部被過濾掉。成立後把



（上鎖）、把



並且



。

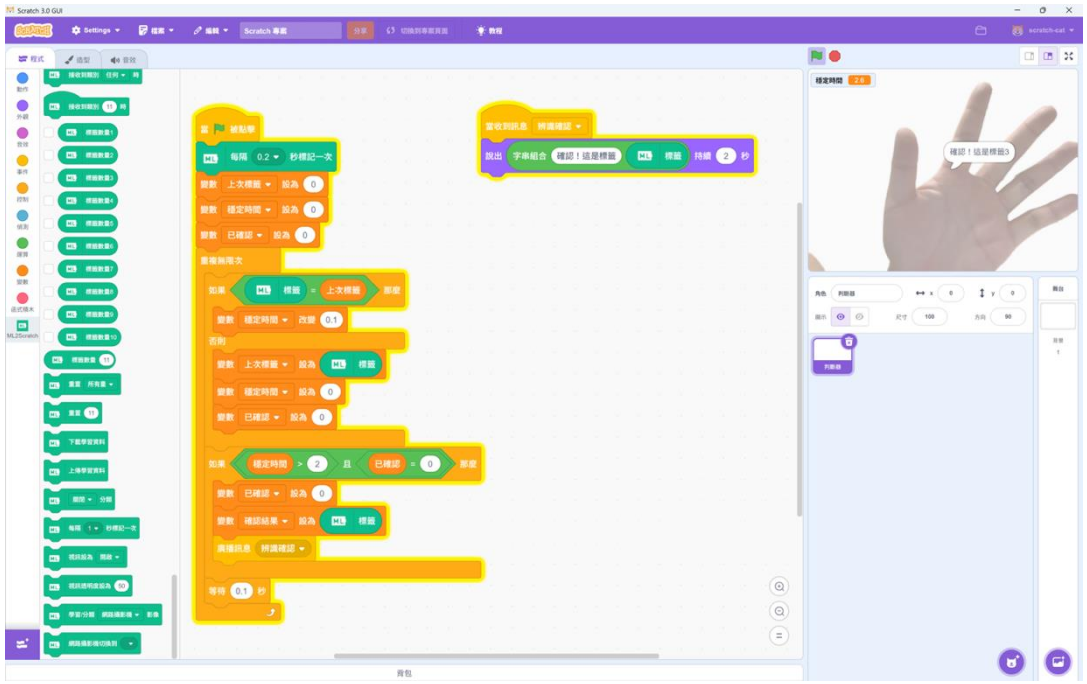
- 「已確認」的上鎖作用：確認過一次後，即使標籤繼續不變，也不會每 0.1 秒重複觸發；直到標籤改變（回到上一點的「不一樣」分支解鎖）才能再次確認。



- 第二段程式：才是真正執行動作的地方—本例是說出確認結果；替換這裡的內容，就能把整套機制套用到猜拳對戰、垃圾分類等任何作品。

- 門檻 2 秒可調整：越長越不易誤判、但反應越慢，這是「準確 vs. 即時」的取捨。

5. 完成後即可進行測試，可嘗試同個手勢持續 2 秒，系統是否能持續 2 秒後判斷出正確的標籤。

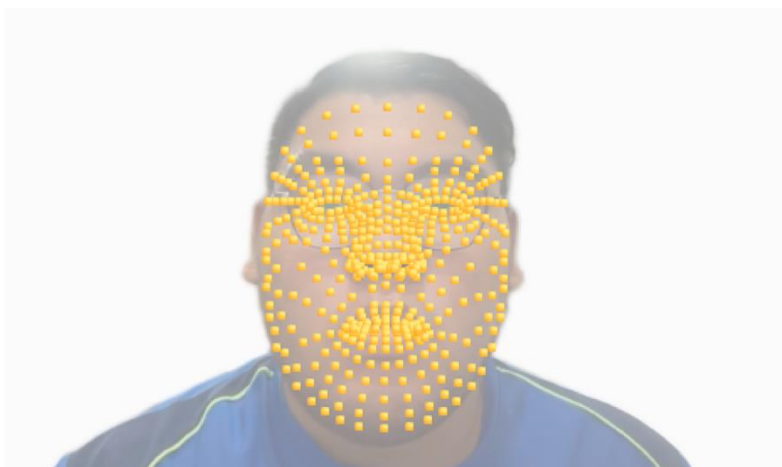
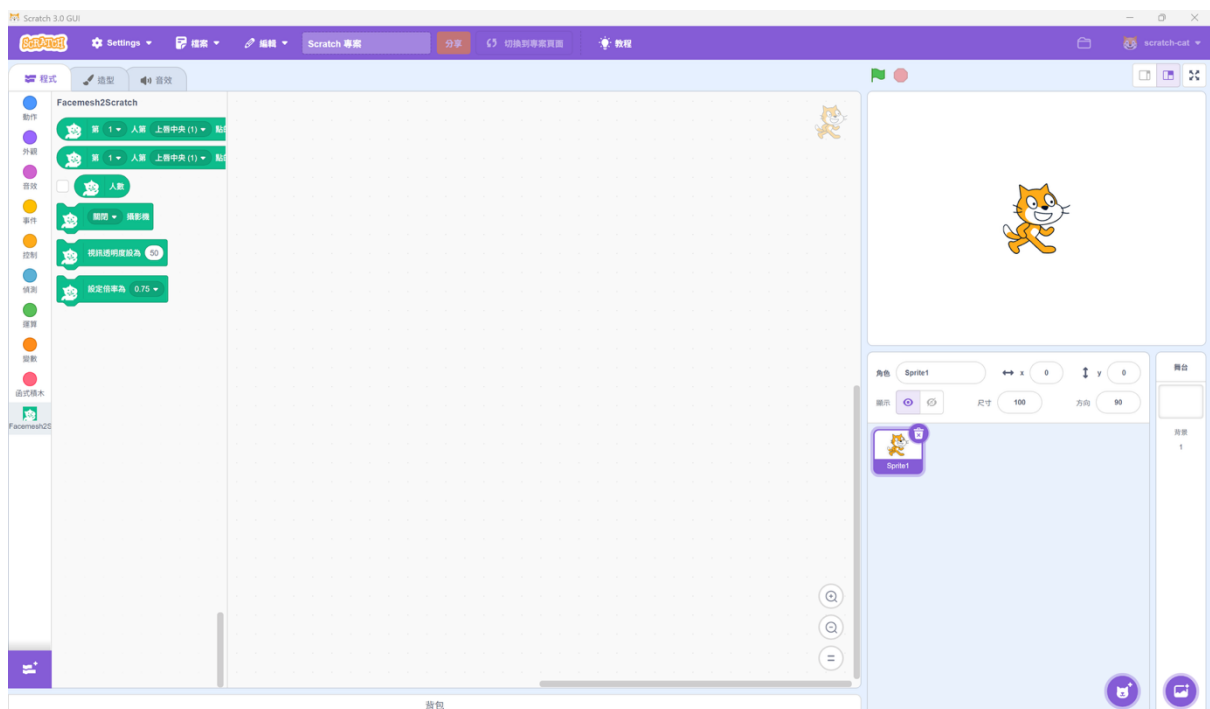


第三章 Facemesh2Scratch：臉部網格偵測模組

3.1 功能介紹

Facemesh2Scratch 使用 Google MediaPipe 的 Facemesh 模型，只靠一般攝影機就能即時偵測臉部的關鍵點（形成一張覆蓋全臉的「網格」），並且最多可同時追蹤 10 張臉。它不需要任何訓練，載入後立刻可用。

每個關鍵點都有編號（1~468）與舞台座標（x、y），您可以取得任何一點的位置，做出例如：虛擬眼鏡或鬍子貼紙、用頭部移動控制遊戲、偵測張嘴／眨眼的體感互動、多人同樂的臉部遊戲等。



實用計算：拿到關鍵點座標後，可以算什麼？

每個關鍵點都會回報舞台座標 x（-240~240）與 y（-180~180）。只要簡單的加減乘除，就

能做出各種偵測（以下公式 Facemesh 與 Handpose 通用，所需的減法、絕對值、平方根、atan 等都在「運算」分類）：

- **開口／張閉程度**：開度 = (上面點的 y) - (下面點的 y)。例：張嘴偵測就是 y (14 上唇) - y (15 下唇)，數值越大嘴張越開，超過門檻就觸發。
- **兩點靠得近不近**：分別算 $|x_1 - x_2|$ 和 $|y_1 - y_2|$ ，兩個若都小於門檻，則表示兩點靠很近。用途：捏合、碰觸判斷。
- **兩點的直線距離**：距離 = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。適合需要「真正距離」的作品。
- **傾斜角度**：角度 = $atan(\frac{y_1 - y_2}{x_1 - x_2})$ ，數學函數選「atan」。例：拿左右眼外角（編號 264 與 34）代入，就能算出頭歪了幾度，可以做「歪頭控制方向」的遊戲。
- **在畫面的哪一邊**：x < 0 在舞台左半邊、x > 0 在右半邊；y 同理分上下。用途：「頭移到左邊就往左走」這類體感控制。

小提醒：所有門檻數值都不是標準答案，每個人、每個環境都不同，請用「勾選監視器觀察數值 → 再決定門檻」的校準流程來取得。

3.2 積木一覽

本模組共 6 個積木。原版積木文字為英文，下表提供中文對照：

積木	類型	功能說明
	功能型積木	第 N 個人 (1~10) 的第 K 號關鍵點 (1~468) 的 x 座標。
	功能型積木	第 N 個人 (1~10) 的第 K 號關鍵點 (1~468) 的 y 座標。
	功能型積木	目前畫面中偵測到的人臉數量 (0~10)。可用來做多人互動或「有沒有人」的判斷。
	指令型積木	控制攝影機畫面：關閉／開啟／翻轉（左右鏡像，如照鏡子效果）。
	指令型積木	設定攝影機畫面透明度 (0 = 清楚、100 = 隱形)。
	指令型積木	設定座標換算倍率 (0.5 / 0.75 / 1 / 1.5 / 2.0，預設 0.75)。影響關鍵點座標對應到舞台的縮

積木	類型	功能說明
		放比例。

3.3 常用臉部關鍵點對照

下表為本軟體所提供的臉部位置與其關鍵點的編號：

臉部位置	關鍵點編號
鼻尖	2
鼻子下緣	3
右鼻翼	99
左鼻翼	328
右眼外角	34
右眼內角	134
右眼上緣	160
右眼下緣	146
左眼內角	363
左眼外角	264
左眼上緣	387
左眼下緣	375
右眉外側	71
右眉中央	106
右眉內側	108
左眉內側	337
左眉中央	335
左眉外側	301

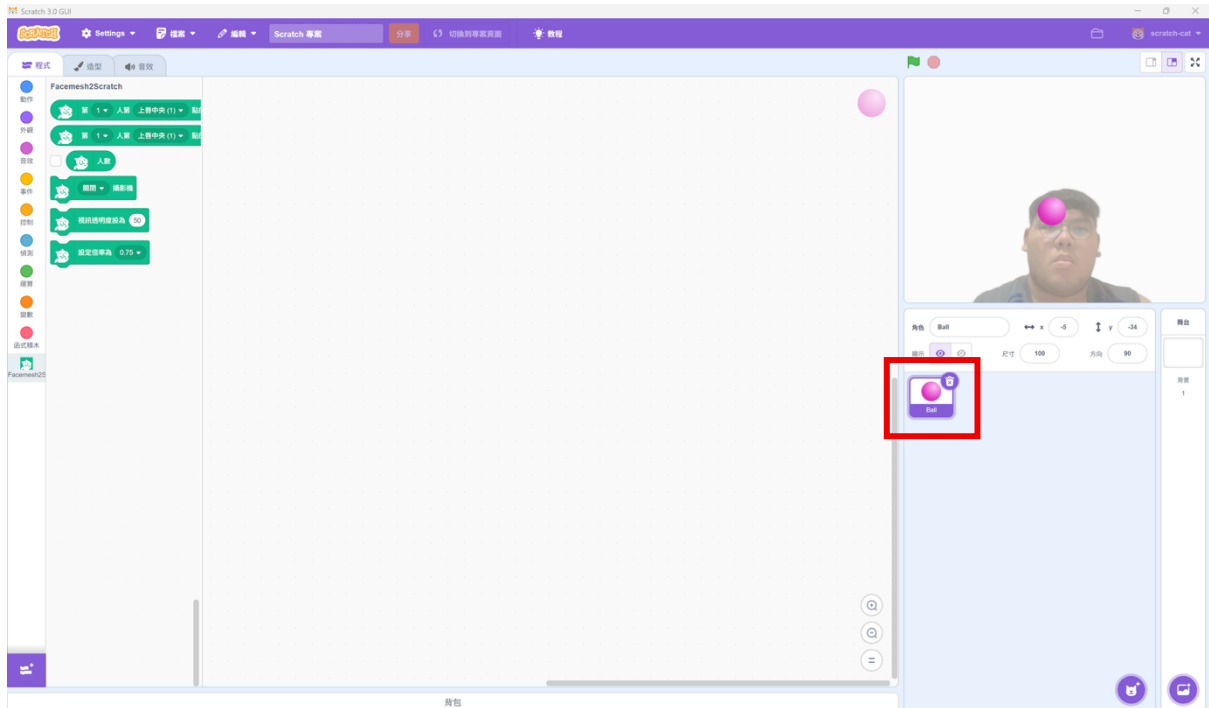
眉心	10
右嘴角	62
左嘴角	292
上唇中央	1
下唇中央	18
上唇內緣中央	14
下唇內緣中央	15
額頭頂端	11
額頭中央	152
下巴	153
右臉頰	206
左臉頰	426
臉部右緣	235
臉部左緣	455

3.4 範例操作：小丑鼻貼紙

目標：讓一顆紅色圓球永遠貼在您的鼻尖上，就像相機 App 的臉部貼紙特效。

操作步驟

1. 載入 Facemesh2Scratch 模組並允許攝影機，等待模型載入完成。
2. 新增一個角色：點右下角「選個角色」，選擇「Ball」（球）。



3. 在該角色上組合以下程式：



► 積木程式詳細說明

- 
 ：每個瞬間都把球移動到「第 1 人第 2 號關鍵點」（約為鼻尖）的最新座標—追蹤效果就是「不停地重新定位」做出來的。
- 
 是回報值，
- 
 分別塞進 之中；只要把關鍵點編號換掉，貼紙就會貼到臉上的其他位置。

4. 按下綠旗，對著鏡頭左右移動、點頭—紅球會緊緊跟著您的鼻尖！



3.5 延伸範例 - 張嘴偵測

上唇中央（關鍵點 14）與下唇中央（關鍵點 15）的 y 座標差距，就是嘴巴張開的程度。閉嘴時差距接近 0，張大嘴時可達 30 以上。我們把這個差距存進變數，超過門檻就觸發效果。

操作步驟

1. 新增一個角色來當「偵測器」。
2. 在「變數」分類點「建立一個變數」，命名為：嘴巴開度。建立後保持勾選，讓它顯示在舞台上。



3. 在該角色上組合以下程式：



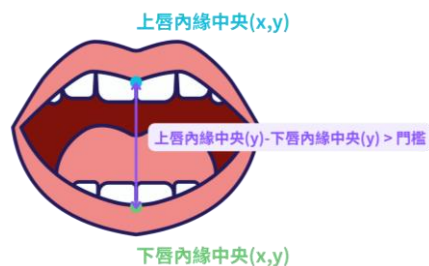
► 積木程式詳細說明

- 核 心 計 算 :



用「運

算」類的減法積木，把上唇內緣中央（14）與下唇內緣中央（15）兩個 y 座標相減，差值就是嘴巴張開的程度（上減下才是正值）。



- 用「判斷」類的積木，判斷差值是否超過門檻：差值超過門檻視為「張嘴」，觸發音效。門檻 15 不是固定答案，必須經過校準（見下方步驟）。
 - 整段放在「重複無限次」中，代表每個瞬間都重新量測、判斷——這就是「即時偵測」的程式寫法。
4. 校準門檻：按下綠旗，看著舞台上的「嘴巴開度」之數值——記下閉嘴與張大嘴時的數值，把程式中的門檻改成居中偏低的值。



5. 測試效果：閉嘴時應該安靜，一張嘴就播放音效。若太靈敏，則把門檻調大；若沒反應則調小，或靠近鏡頭一點再重新校準。

校準 (calibration)

每個人臉型、坐姿距離不同，張嘴的數值也不同，所以門檻不能寫死、必須實測決定——這正是所有感測器應用（體重計歸零、搖桿置中）都要做的「校準」。

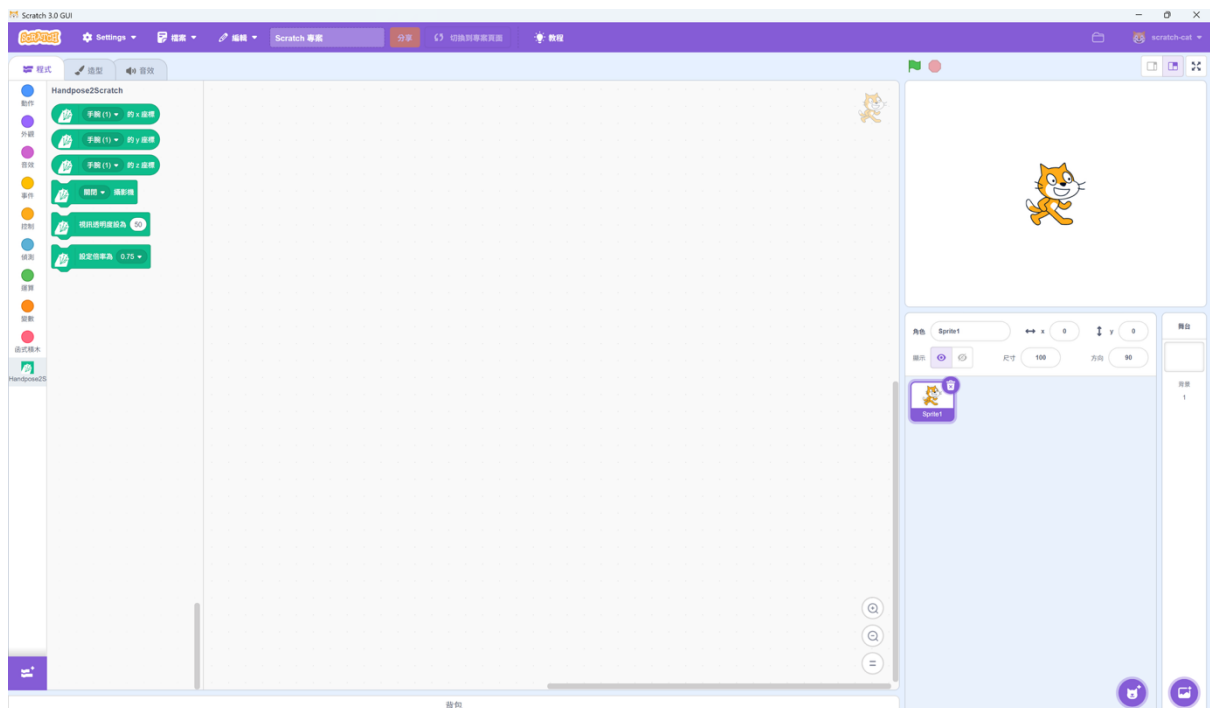
第四章 Handpose2Scratch：手部姿勢偵測模組

4.1 功能介紹

Handpose2Scratch 使用 MediaPipe 的 Handpose 模型，透過攝影機即時偵測「一隻手」的 21 個關節點（手腕 1 點＋每根手指 4 點），回報每個點的 x、y、z 座標。與 Facemesh 一樣不需訓練、載入即用。

有了手部關鍵點，您的手就變成了控制器：用食指在空中畫圖、用手掌位置控制角色移動、偵測手指捏合當作「點擊」、比出不同手勢觸發不同效果等。

特別的是，本模組還提供 z 座標（深度），代表關鍵點距離鏡頭的遠近變化，可以做出「手往前推」的推拉互動。



實用計算：手部座標的常用算法

21 個關節點同樣回報舞台座標，第三章的距離、角度公式全部通用。以下是手部特別常用的幾招：

- **手掌控制角色**：最簡單也最穩：直接把  塞進  中，角色就左右跟著手移動。
- **捏合（手勢點擊）**：拇指尖（thumb）與食指尖（index finger）的 x 差、y 差取絕對值若都小於 30，則代表捏合。門檻需依實測而有所調整。

- **手指有沒有伸直（簡單判斷）**：手掌正面朝鏡頭、手指朝上時：指尖的 y 若比同一根手指第 3 關節（根部）的 y 高，則表示伸直；若低，則代表彎曲收起。五根都判斷一次，就能做出「數手指」的作品。
- **手往前推**： z 是「相對深度」：不是公分數，但變化方向可靠。記錄「現在的 z - 剛才的 z 」，短時間內明顯變化，則代表手有往前推／往後收。
- **手的傾斜角度**： $\text{atan}(\frac{y_1-y_2}{x_1-x_2})$ ，代入手腕與中指根部兩點，可判斷手掌左右傾斜，做「方向盤」遊戲。

4.2 積木一覽

本模組共 6 個積木。原版積木文字為英文，下表提供中文對照：

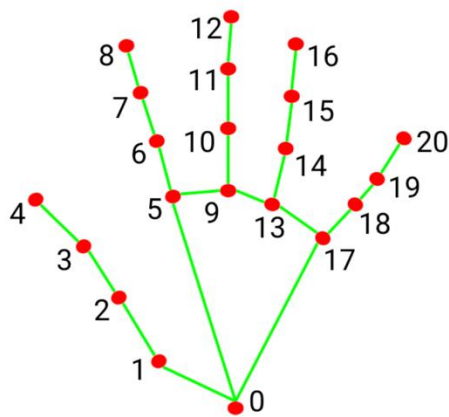
積木	類型	功能說明
	功能型積木	指定手部關節點的 x 座標。
	功能型積木	指定關節點的 y 座標。
	功能型積木	指定關節點的 z 座標（深度，離鏡頭越近／越遠數值會變化，適合做前後推拉偵測）。
	指令型積木	控制攝影機畫面：關閉／開啟／翻轉（左右鏡像，如照鏡子效果）。
	指令型積木	設定攝影機畫面透明度（0 = 清楚、100 = 隱形）。
	指令型積木	設定座標換算倍率（0.5／0.75／1／1.5／2.0，預設 0.75）。影響關鍵點座標對應到舞台的縮放比例。

4.3 21 個手部關節點對照表

選單依序為以下 21 個部位。每根手指從根部到指尖各有 4 個點：

部位名稱	關鍵點編號
手腕	1

部位名稱	關鍵點編號
拇指根部	2
拇指第二關節	3
拇指第一關節	4
拇指尖	5
食指第三關節	6
食指第二關節	7
食指第一關節	8
食指尖	9
中指第三關節	10
中指第二關節	11
中指第一關節	12
中指尖	13
無名指第三關節	14
無名指第二關節	15
無名指第一關節	16
無名指尖	17
小指第三關節	18
小指第二關節	19
小指第一關節	20
小指尖	21



- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

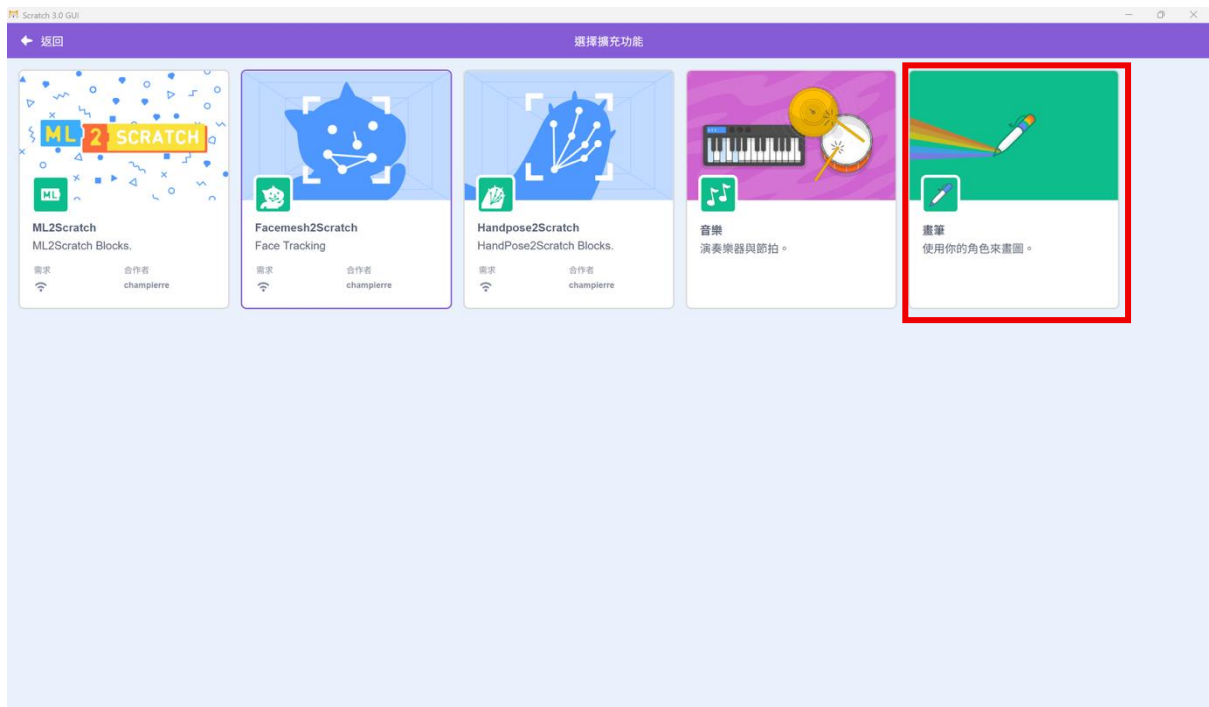
(圖片來源於 Google AI Edge)

4.4 範例操作：食指空中畫筆

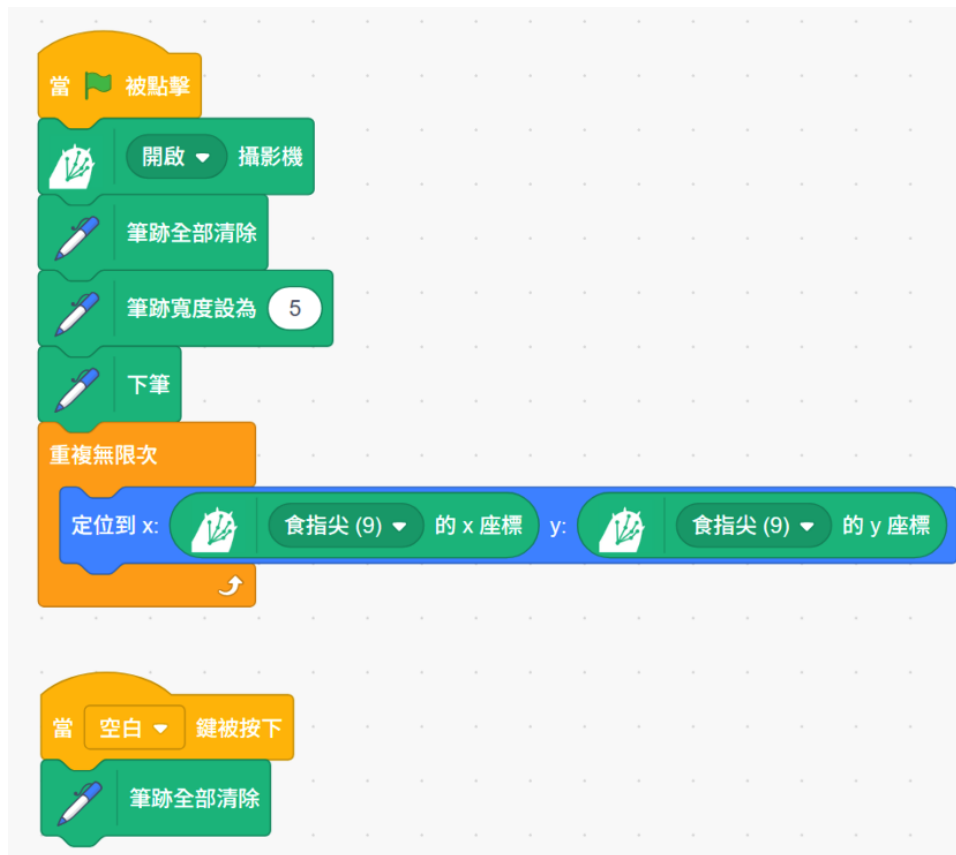
目標：伸出食指，在空中揮動就能於舞台上畫圖，按空白鍵清除重畫。

操作步驟

1. 載入 Handpose2Scratch 模組並允許攝影機，等待模型載入。
2. 加入 Scratch 內建的「畫筆」擴充模組。



3. 選擇任一角色（畫筆從角色的位置畫出，可將角色縮小或設為隱藏）。
4. 在角色上組合以下程式：



► 積木程式詳細說明

- 開頭「筆跡全部清除+下筆」：先清空畫布，然後讓筆保持落下狀態。

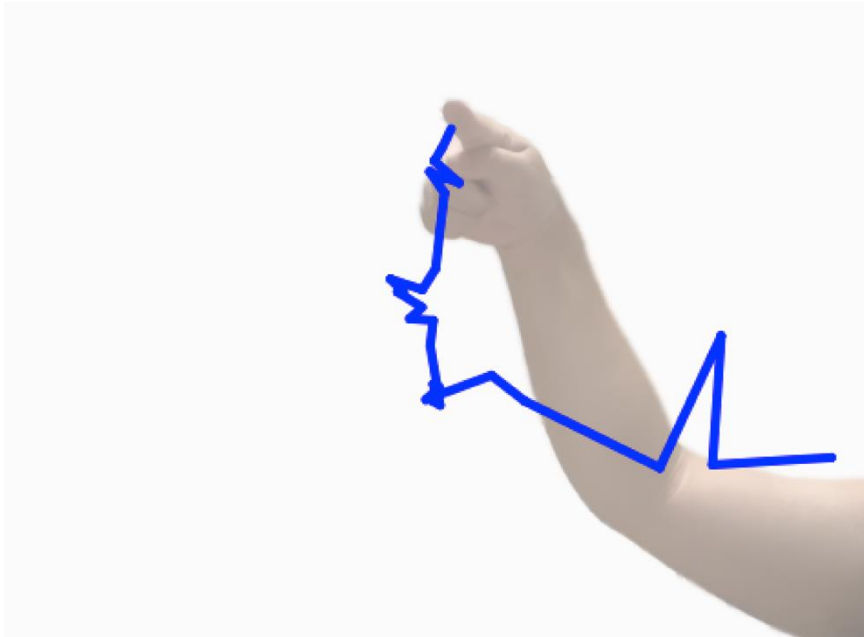


- ：角色每個瞬間都移動到食指指尖的最新位置，因為筆是落下的，移動軌跡就成了畫出的線。



- ：第二段獨立程式，隨時按空白鍵清空重畫。

5. 按下綠旗，把一隻手舉到鏡頭前（距離約 30~80 公分、手掌朝向鏡頭），用食指在空中寫字畫圖吧！



4.5 延伸範例

延伸範例 1：捏合控制下筆

原理：拇指指尖與食指指尖靠得夠近等同於「捏合」。我們分別計算兩點 x 、 y 座標差的絕對值，都小於門檻（例如 30）就視為捏合、執行下筆，否則停筆。

操作步驟

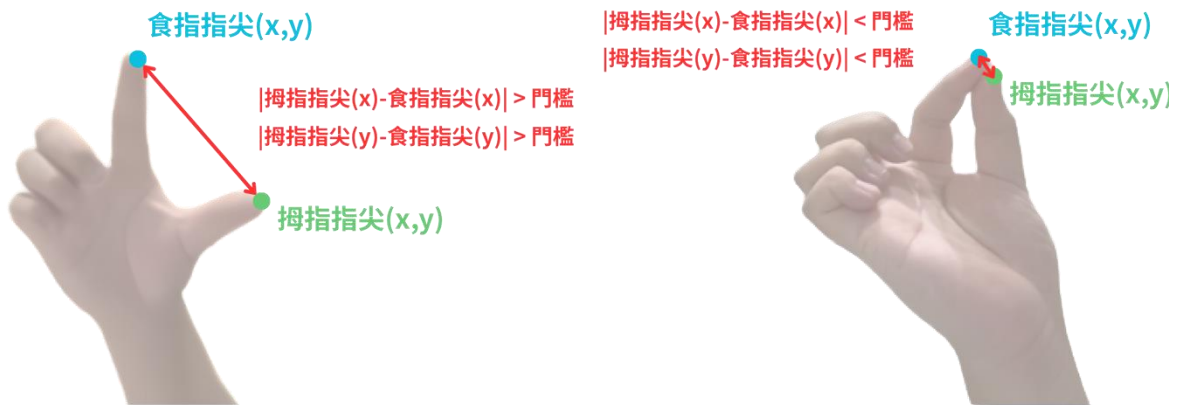
1. 修改主程式：把先前 4.4 專案中程式開頭的「下筆」移除，改成迴圈內的條件判斷，完整程式如下：



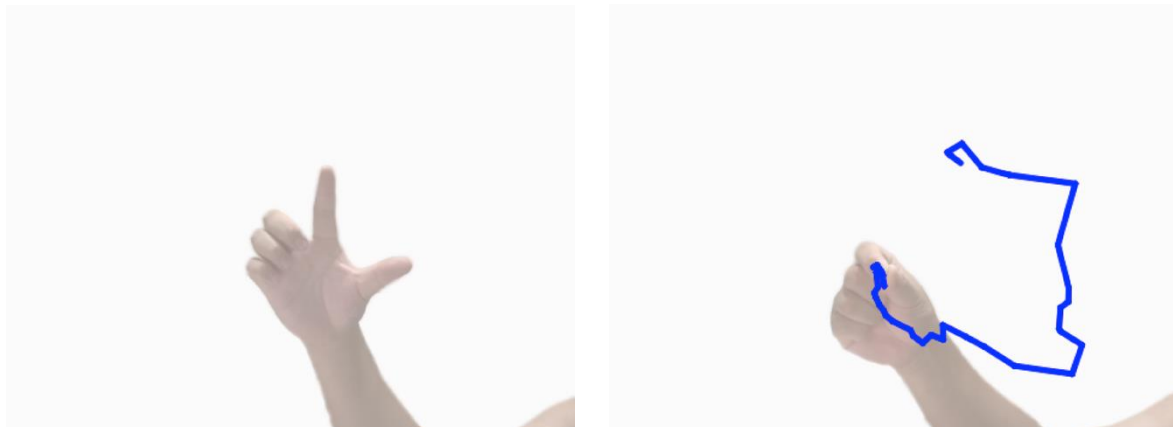
► 積木程式詳細說明



：計算拇指指尖與食指指尖的 x 差值的絕對值與 y 差值的絕對值，在各自判斷小於門檻條件。



- 取絕對值之原因，由於座標相減可能是負數，絕對值把它變成單純的「距離差」，只關心遠近、不關心方向。
 - 判斷邏輯：x 差與 y 差「都」小於 30，代表拇指與食指指尖非常接近，也就表示捏合，所以下筆；若任一超過門檻，則表示手指張開，因此停筆。角色仍持續跟著食指移動，所以張開時是「移動但不畫」。
 - 門檻需校準：若捏了不畫，則需調大門檻；若沒捏就誤畫，則需調小門檻。
2. 測試流程：按下綠旗，手舉到鏡頭前—手指張開時移動角色但不畫線；拇指與食指捏在一起時開始畫線；張開後停止。試著用「捏合寫字、張開換位置」的節奏寫出自己的名字。



3. 校準門檻：若捏合了卻不畫，把門檻值調大；若手指沒捏就誤畫，則調小。想精準校準，可建立變數「指距 X」設為那個絕對值，勾選顯示在舞台上，觀察捏合與張開時的實際數值再決定門檻。

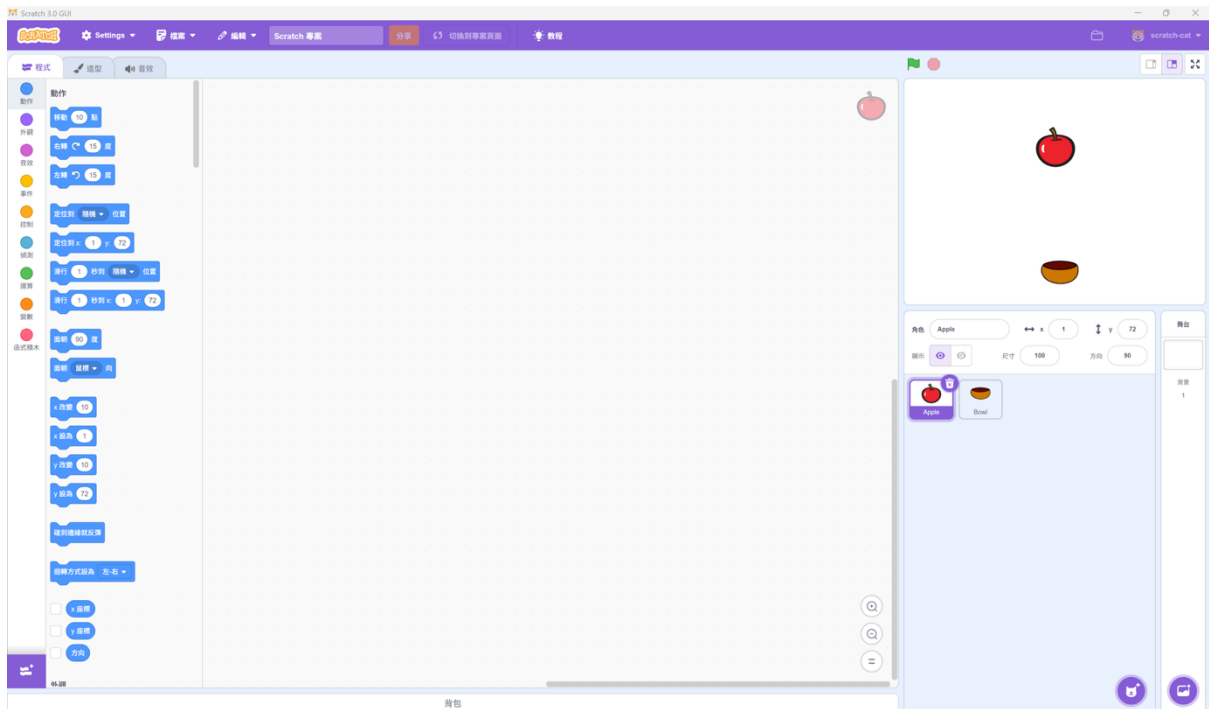
延伸範例 2：手掌接水果

原理：用手腕的 x 座標控制籃子左右移動，接住從天上掉下來的水果。這是把「手勢控制」套

進經典遊戲架構的範例，完成後可自行替換成任何「控制角色移動」的遊戲。

操作步驟

1. 開新專案，載入 Handpose2Scratch 模組。點右下角「選個角色」，新增兩個角色：籃子與蘋果。



2. 在「變數」分類點「建立一個變數」，命名為：分數。保持勾選，讓分數顯示在舞台上。

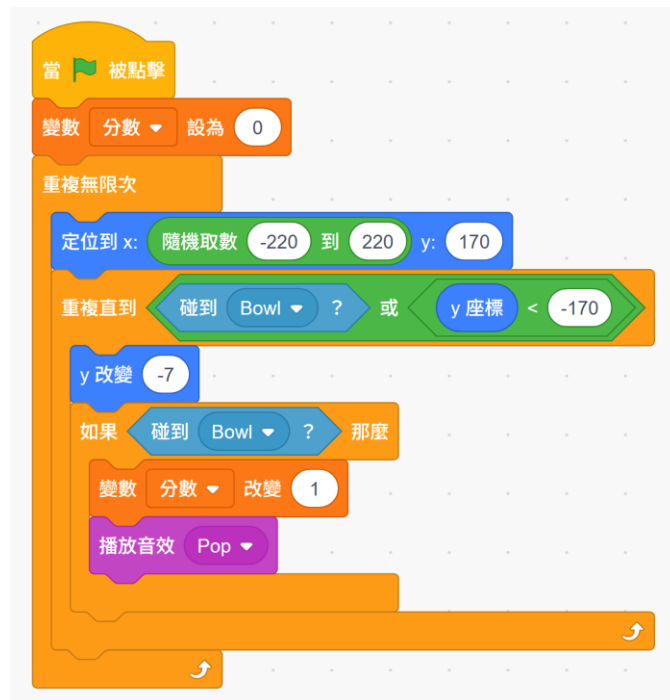


3. 點選籃子角色，組合籃子程式—只做一件事：x 座標永遠跟著手腕：



► 積木程式詳細說明

- ：開始時把籃子固定在畫面下方中央。
 - ：只改 x、不改 y—籃子高度不變，左右位置永遠跟著手腕。搭配鏡像視訊，手往右籃子就往右，就像手裡真的拿著籃子。
4. 點選蘋果角色，組合蘋果程式—從上方隨機位置掉落，接到加分、落地重來：



► 積木程式詳細說明

- 最外層 ：讓遊戲一輪接一輪永遠進行。
- 每輪開始 ：蘋果回到舞台頂端的隨機位置，玩家無法預測落點。
-  內執行 ：蘋果每次下移 7 步，代表持續下落；若碰到籃子或掉出畫面底部就結束下落。



- 落地判定：——只有「接到」才加分；沒接到就直接回到迴圈開頭重新掉落。
5. 測試流程：按下綠旗，把手舉到鏡頭前，左右移動手，籃子跟著移動；蘋果從頂端掉下，接到時「分數+1」並發出音效，沒接到則落地後回到頂端重來。

