

# Scratch！多媒體遊戲設計

我錯了！大多數從事 Scratch 教學、著作的老師都錯了！Scratch 並不是程式設計工具，國小、國中、高中學生也不應該學程式設計，當全世界頂尖國家都將程式設計課程提早到小學的當下，我的說法是否不負責任、甚至無知！

台灣的少棒、青少棒、青棒揚名國際，台灣職棒卻連世界的尾巴的沾不上！Why...Why...Why...？（連 3 壞），在美國多數家庭、小孩對於 Sports 的態度是：Play、Game For Fun！但在台灣卻完全不是那麼一回事！它變成宣揚家威、校威、國威的工具，因此少棒選手從小打球就是為了名次，棒球不再是一種運動、不再帶來快樂！揠苗助長的結果，小選手的運動生涯提前報廢了！

程式設計也是一樣，偉大的教育專家們又是那一套「中學為體、西學為用」，引進先進國家的教育體制，卻無視國內畸形教學生態，Scratch 又被當成程式設計語言在教，還在介紹功能、指令、邏輯！

Scratch 的出現讓我雀躍！原來「程式設計」可以是那麼有趣！以遊戲設計、快樂學習為出發點，不必管指令、邏輯、直覺的拼圖組裝，回到學徒教育的初衷：「做中學」！5 分鐘就有成果，一個個令人有成就感的「動作」，再組合為一個「小作品」，但如果還是那一套台式教育思維，將學習當成一種目的，違反 Scratch 基本精神，那 Scratch 對於台灣學生而言仍是一場災難！

台灣商管教材研發學會

林文恭、吳進北

2018/10

**Scratch !**

它不是程式設計工具

它是遊戲設計平台！

程式設計是苦悶的！

遊戲設計是愉悅的！

這是一本以 **Scratch** 為工具  
介紹遊戲設計的書  
結合 **TELLO** 無人機  
讓學習樂趣飛上天

全書教學影片

[gogo123.com.tw/?page\\_id=10045](http://gogo123.com.tw/?page_id=10045)



## 目錄

第一篇：認識 <b>Scratch</b> .....	003
Scratch 簡介 .....	004
專案 01：水中的魚 .....	012
專案 02：水中的氣泡 .....	026
專案 03：吃角子老虎機 .....	038
專案 04：500 小魚 .....	046
專案 05：幾何小畫家 .....	054
第二篇：系統化設計 .....	069
專案 06：小小音樂家 .....	070
專案 07：小精靈吃蘋果 .....	080
專案 08：打磚塊 .....	096
專案 09：填充方塊遊戲 .....	106
專案 10：隔空打蝙蝠 .....	122
專案 11：物流配送模擬 .....	136
第三篇：結合 <b>TELLO</b> 無人機 .....	161
專案 12：TELLO 初體驗 .....	162
專案 13：無人機物流配送 ...	176

# 幾何小畫家

## 專案企劃

### 發想

■ 小時候的玩具：萬花尺（右圖）

- ▶ 尺的中央為一圓型洞
- ▶ 洞的周邊為尺輪之牙
- ▶ 中間置入一齒輪  
齒輪上有許多小圓孔

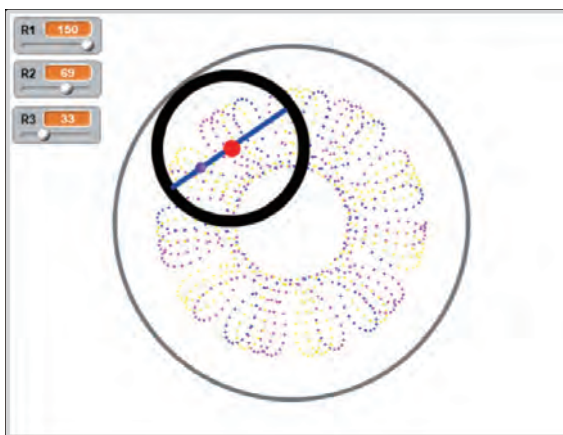


■ 將筆插入小圓孔，繞著圈圈移動便可畫出漂亮幾何圖形（右下圖）。

繪出的幾何圖樣取決於 3 個要素：

- R1：圓洞半徑
- R2：齒輪半徑
- R3：筆孔距離齒輪圓心距離

繪圖原理非常簡單，就是三角函數  $\text{Sin}()$ 、 $\text{Cos}()$  的應用而已。



本專案設計的主要目的是介紹 Scratch 簡單易用的繪圖功能。

## 建立舞台

本專案使用系統預設白底舞台。

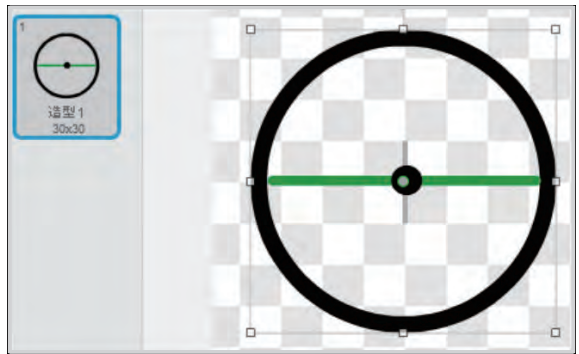
## 建立角色

- 畫大圓：由範例庫取得（Pencil）  
調整大小：30×30  
中心位置：筆尖
- 筆心：自行繪製紅色實心圓  
調整大小：10×10  
中心位置：圓心



齒輪：自行繪製，構造如下：

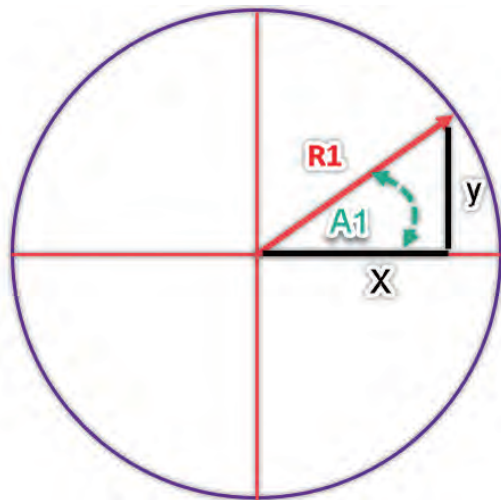
- 圖片大小：30×30
- 一個黑色粗外框圓形
- 一條穿越圓心綠色直線
- 一個位於圓心的黑色實心點



## 畫大圓

圓周的軌跡公式：

- $X = R1 \times \text{Cos}(A1)$
- $Y = R1 \times \text{Sin}(A1)$



## 筆心位置

### ◎ 筆心移動軌跡分析

齒輪上的筆心是繞著齒輪中心轉動

$$\begin{aligned} \rightarrow \text{筆心移動軌跡 } X3 &= \text{齒輪中心移動軌跡} + \text{齒輪 } R3 \text{ 半徑自轉軌跡} \\ &= X2 + R3 \times \text{Cos}(A2) \\ Y2 &= Y2 + R3 \times \text{Sin}(A2) \end{aligned}$$

(A1：繞圓轉動角度、A2：齒輪自轉角度)

### ◎ 動作 -1：筆心位置起始設定

#### 1 設定：

系統開始時→隱藏筆心位置

畫圓完畢時→顯示筆心位置



#### 2 設定筆心：

筆跡寬度：2

起始位置座標



**說明** 筆心位置的繪圖我們採用較小的點，因此設定：2

### ◎ 動作 -2：筆心移動、畫點

齒輪轉動的同時，齒輪中的筆心也跟著轉動，因此我們設計：

- 齒輪每轉動一下就對筆心發出訊息：讓筆心也跟著轉一下、畫點

#### 1 由齒輪發送訊息給筆心

選取角色：齒輪

插入廣播訊息指令

設定訊息內容：畫點



## 2 設定筆心軌跡

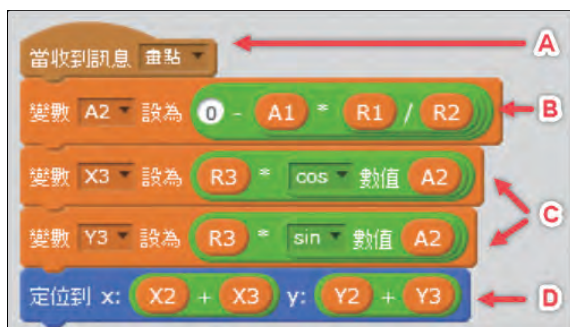
選取角色：筆心

**A** 建立：收到畫點訊息事件

**B** 設定齒輪自轉角度

**C** 設定筆心移動 X、Y 座標

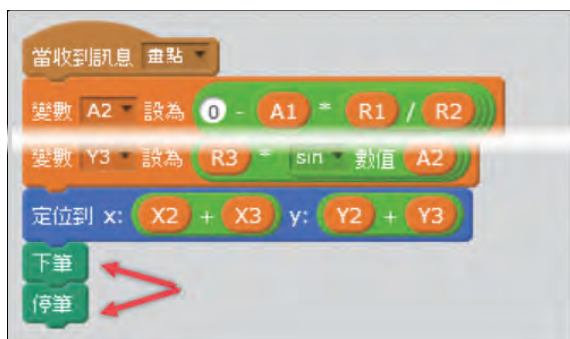
**D** 定位筆心位置



## 3 插入畫點指令

在同一位置：下筆 + 停筆

就是畫一個點



測試：

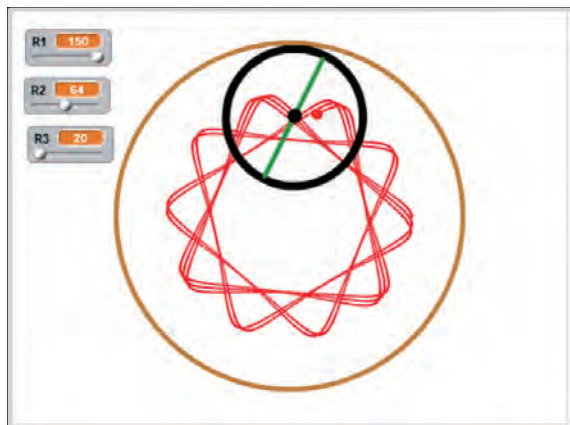
### ■ 執行程式

繪圖結果如右

### ■ 還有 2 個問題尚未克服：

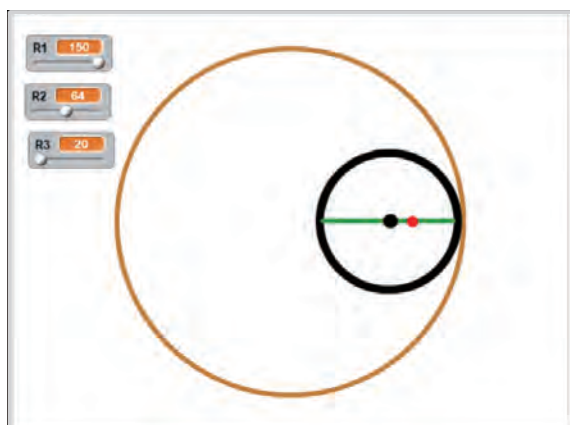
**A** 筆心沒有位於綠色線上

**B** 筆心顏色沒有變化



## 🌀 動作 -3：問題修正 A

由於每一次執行程式齒輪都會轉動，因此在齒輪轉動前應該先讓齒輪面朝右，如此齒輪的綠色線才會與筆心在同一水平。



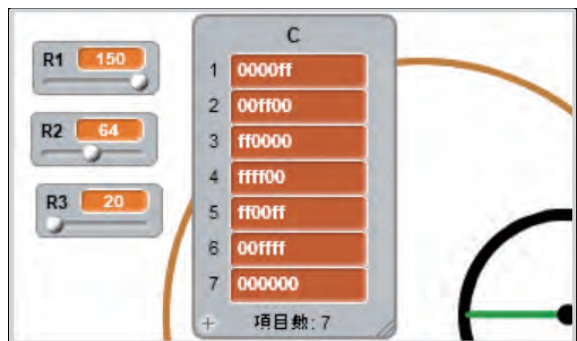
- 選取角色：齒輪
- 插入轉向指令
- 讓齒輪朝右



### ◎ 動作 -3：問題修正 B

我們希望每一圈的顏色都可以有變化，因此使用【隨意取數】。想要以指令控制顏色，就必須使用 16 進位色碼。

- 1 建立色碼清單：C  
新增資料為：7 筆  
依序輸入 7 筆資料  
如右圖：



**說明** 16 進位色碼規則：(以第一筆資料為例)

- A. 總共 6 碼，2 碼一個單元總共 3 個單元，00-00-ff，分別代表：紅 - 綠 - 藍
- B. 00：無，ff 表：有，因此 00-00-ff = 無 - 無 - 藍 = 藍色
- C. 根據 B 點類推，2~7 筆資料分析如下：
 

2：00-ff-00 = 無 - 綠 - 無 = 綠	3：ff-00-00 = 紅 - 無 - 無 = 紅
4：ff-ff-00 = 紅 - 綠 - 無 = 黃	5：ff-00-ff = 紅 - 無 - 藍 = 紫
6：00-ff-ff = 無 - 綠 - 藍 = 青	7：00-00-00 = 紅 - 綠 - 藍 = 白

- 2 新增變數：color
- 3 選取角色：齒輪，插入變數設定指令如下圖：



**說明** 顏色設定的必須是文字串，我們採取 16 進位值，因此開頭必須加上 "0x"。  
例如："0xff0000" 就代表紅色、"0x0000ff" 就代表藍色。

所以我們使用字串組合功能串接 "0x"、C 清單中的 16 進位值。

清單選項中又使用【隨機】，因此可產生隨機取數功能。

#### 4 選取角色：筆心

插入筆跡顏色指令

顏色值：color

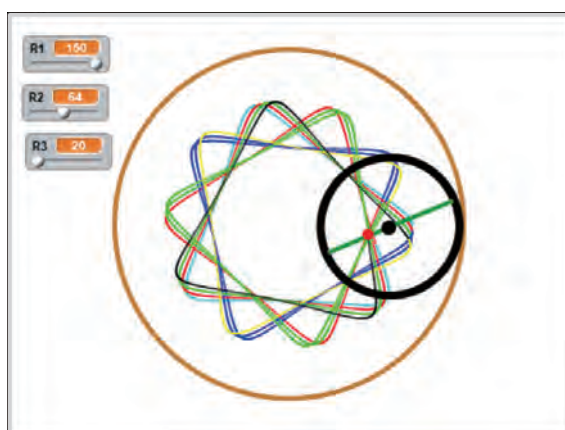


測試：

#### ■ 執行繪圖遊戲

繪圖結果如右

- 完成繪圖後，看著齒輪實在很礙眼！想把完整的圖案 copy 下來就顯得很麻煩，因此們希望可以利用按鍵來切換齒輪的顯示、隱藏。



### ◎ 動作 -4：顯示 / 隱藏齒輪

#### 1 選取角色：齒輪

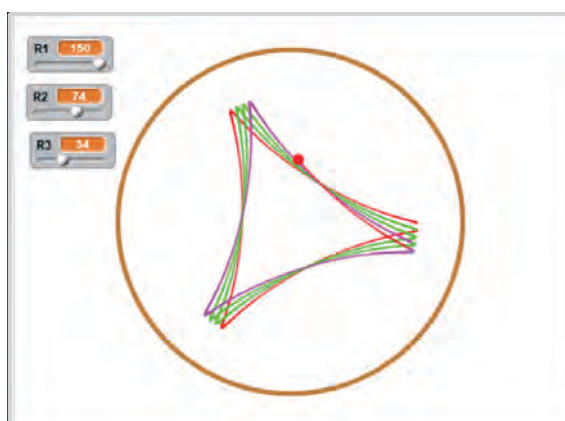
#### 2 建立程式，如右圖



測試：

#### ■ 執行繪圖遊戲

按 0 鍵→齒輪隱藏

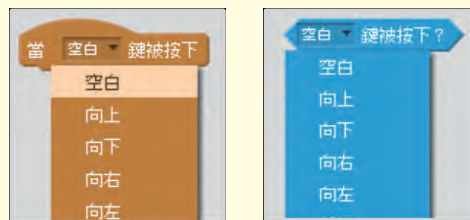




**功能說明 24：偵測按鍵**

以按鍵〔0〕、〔1〕作為齒輪【隱藏 / 顯示】控制鍵。

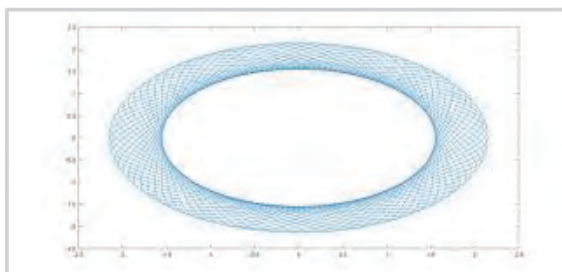
偵測某一按鍵是否被按下有 2 個工具，如右圖：

**專案命名**

- Untitled- x 的專案名稱，更改為：05- 幾何藝術家

**發揮創意**

- 將繪圖板的大圓改為橢圓。



# 3 PART

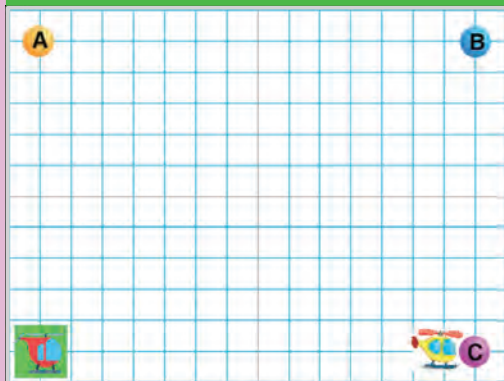
## 結合 TELLO 無人機

### » 程式操控無人機

#### » TELLO 無人機



#### » 物流配送模擬



#### » 無人機飛行程式碼

```
當角色被點擊  
設置速度 110  
起飛  
等待 1 秒  
向上 距離 50  
等待 3 秒  
變數 P 設為 1  
重複 4 次  
  順時針轉 清單第 P 項項目 ( f-A )  
  等待 5 秒  
  向前 距離 清單第 P 項項目 ( f-L )  
  等待 清單第 P 項項目 ( f-L ) / 30 秒  
  變數 P 改變 1  
翻滾 方向 f  
等待 2 秒  
翻滾 方向 b  
等待 2 秒  
降落
```

## 在 Scratch 匯入 Tello 積木

使用不同的工具操控 TELLO 飛行，有以下不同模式：

- A 電腦：以程式控制
- B 手機：以 APP 手機介面控制
- C 手機 + 電玩搖桿 + VR 眼罩：手機搖桿控制

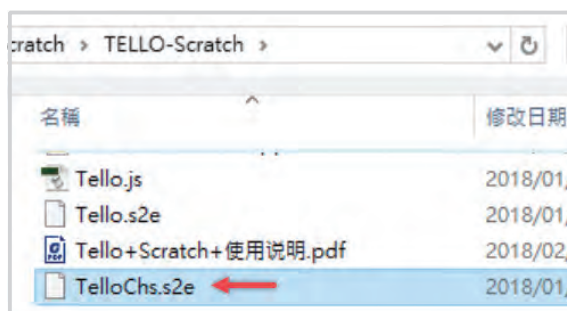
本書重點為【程式控制】無人機飛行，搭配無人機物流配送點規劃的主題，因此焦點聚集在【使用 Scratch 控制 TELLO】！

TELLO 設計了一套飛行積木供 Scratch 使用，這些積木的使用方法與 Scratch 內建的積木完全一樣，可以搭配使用，因為 TELLO 積木是外來物，因此必須先匯入到 Scratch 專案內才能使用。

- 1 在 Scratch 建立新方案
- 2 按住 Shift 鍵不放  
點選：檔案下拉鈕  
點選：匯入實驗性 HTTP 擴充功能



- 3 選取光碟片目錄：TELLO\Scratch  
選取：TelloChs.s2e



**說明** 積木有 2 個版本：Tello.s2e → 英文版、Tello.Chs.s2e → 中文版。

- 4 點選：程式→更多積木  
看到 TELLO 積木成功匯入  
如右圖：



- 完整飛行積木共 12 個，如下圖：



## ◎ 以鍵盤控制 TELLO

TELLO 提供 12 個積木，其中 11 個飛行動作（空翻：前空翻、後空翻），1 個設定動作，下方我們就針對 11+1 個動作規劃鍵盤對照圖，為了使用者操作便利考量，我們只使用：數字鍵、方向鍵，規劃如下：

數字鍵：：1~9 (5 省略不用)

7 上升	8 起飛	9 前空翻
4 左旋	5	6 右旋
1 下降	2 降落	3 後空翻

- 建立指令如下圖：



方向鍵：上、下、左、右



- 建立指令如下圖：



**說明** 指令中的參數解釋如下：

距離：單位為公分，最大值 500 = 500 公分 = 5 公尺。

注意！若飛行距離為 10 公尺，就必須下 2 次飛行 5 公尺的指令。

旋轉：角度，一個圓周：360 度。

翻轉：b → Backward 後空翻，f → forward 前空翻。

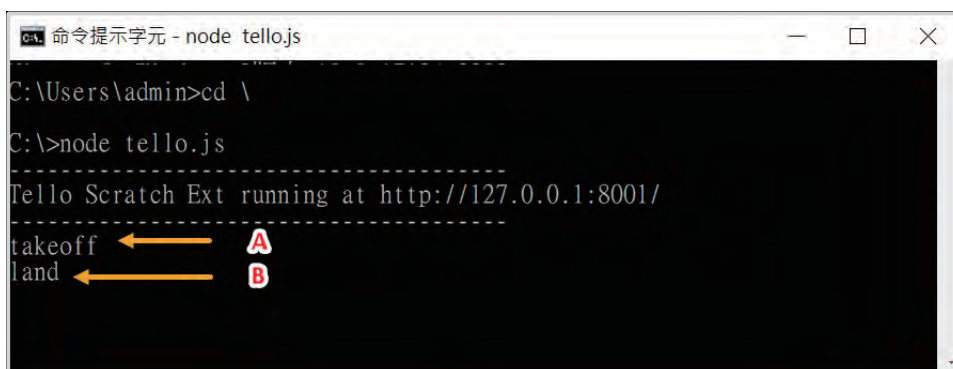
測試：

- 按 8 鍵 → TELLO 垂直飛起
- 按 2 鍵 → TELLO 垂直下降
- 切換到命令提示字元視窗

看到系統對 TELLO 下了 2 列指令

**A** takeoff (起飛)

**B** land (著陸)



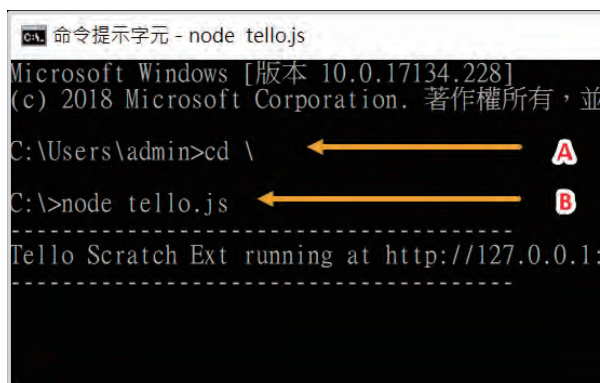
```
C:\Users\admin>cd \  
C:\>node tello.js  
-----  
Tello Scratch Ext running at http://127.0.0.1:8001/  
-----  
takeoff  
land
```

## 程式控制 TELLO 初體驗

TELLO 提供給 Scratch 的積木 (動作) 只有 13 個，目前還缺少拍照、攝影的動作有些遺憾！這一節我們就要使用 Scratch 來設計 TELLO 的飛行程式。

### ◎ 建立連線

- 1 切換到字元命令模式
- 2 執行指令：
  - A** cd \
  - B** node tello.js



```
Microsoft Windows [版本 10.0.17134.228]  
(c) 2018 Microsoft Corporation. 著作權所有，並  
C:\Users\admin>cd \  
C:\>node tello.js  
-----  
Tello Scratch Ext running at http://127.0.0.1:  
-----
```

3 開啟 TELLO 電源

4 建立 WIFI 連線



### ◎ 設計飛行路徑、動作



**說明** 設計飛行動作時請特別注意！

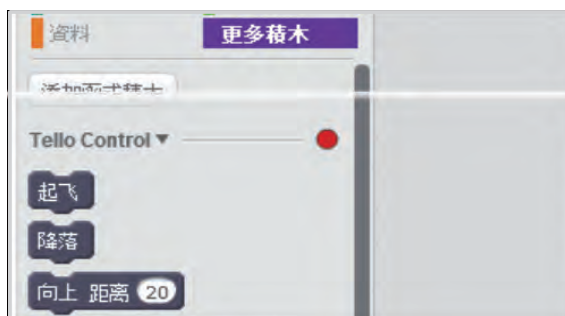
每個動作必須確實完成後，才能進行下一個動作，例如：

第一個動作：【起飛】，假設【起飛】動作需要 3 秒鐘才能完成，我們下指令時就必須在【起飛】指令後，加上一個【等待：3 秒】的指令，才能往下作第 2 個動作【前飛：300】。

如果【飛行動作】之間沒有足夠的等待時間，中間的步驟將會被直接略過！

## ◎ 建立專案

- 1 建立新專案
  - 2 匯入 TELLO 積木  
執行：Tello.Chs.s2e
- 匯入 TELLO 積木如右圖：



**說明** TELLO 無人機是大陸大疆公司研發的產品，因此積木的中文用的是簡體版。

- 3 根據飛行路徑設計  
建立飛行程式
- 4 在每一個飛行動作之後  
插入等待指令  
如右圖：



測試：

- 觀察每一個飛行是否正確執行，若某些飛行動作被略過，表示等待時間不足，必須增加等待秒數！