The background features several horizontal, overlapping brushstrokes in a vibrant teal color. A white, thick-lined frame is superimposed over the center, enclosing the text. The text is white and bold, centered within the frame.

**AIOT Coding**  
**智慧物聯-**  
**Scratch Fun**

# [ 目錄 part1 ]



- 01 yabboni-介紹
- 02 yabboni-感測參數介紹
- 03 yabboni-操作功能介紹
- 04 yabboni-配件介紹
- 05 yabboni-軸向定義
- 06 yabboni Scratch 連線
- 07 yabboni - Scratch UI介紹
- 08 yabboni-USB連線
- 09 yabboni-藍芽BLE 連線
- 10 yabboni-Scratch連線
- 11 yabboni-Scratch 範例程式

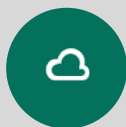
# rabinieri-介紹



rabinieri內建六軸重力感測器 (IMU: Initial Measurement Unit)、BLE藍芽傳輸及運算元件



可即時傳輸感測讀值並提供取樣頻率及動態範圍之多樣選擇



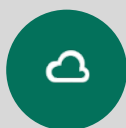
配有LED燈，指示rabinieri運作狀態及電量顯示。



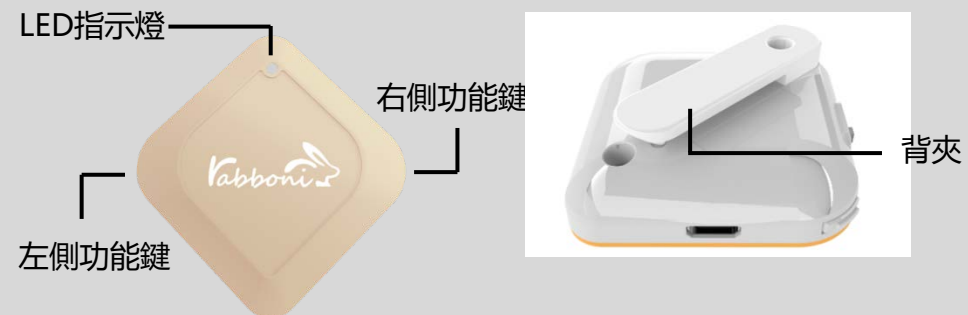
rabinieri 提供Android感測訊號擷取 APP及各式程式教育應用 API



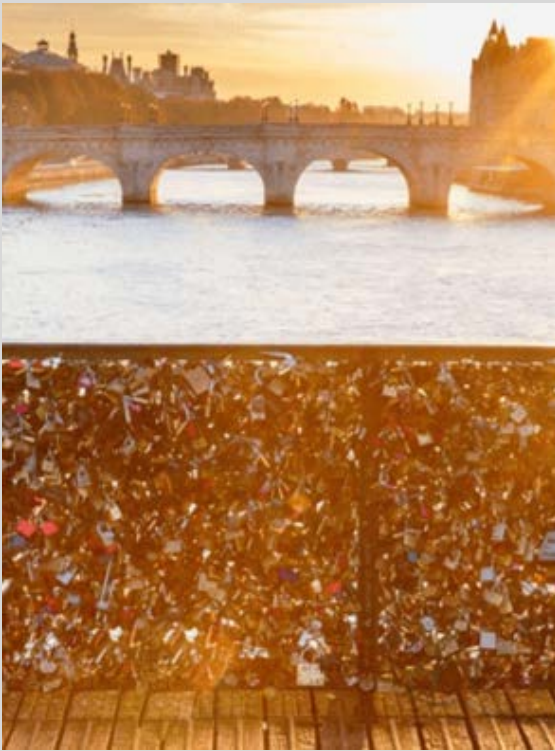
Scratch, Python, Unity, Java, App Inventor



專為 AIoT 程式教育、APP開發、AI 智慧感測互聯或各種智慧化應用之動作偵測相關研究開發使用。



# yabboni-感測參數介紹



Gyro Full Scale Range	Gyro Sensitivity	Accel Full Scale Range
(°/sec)	(LSB/°/sec)	(g)
±250	65.5	±2
±500	32.8	±4
±1000	16.4	±8
±2000	8.2	±16

電池容量	120mAh 鋰離子充電電池
充電方式	USB mini 充電
無線傳輸	Bluetooth 4.0 BLE
充電時間	30分鐘
待機時間	5天 (電源開關鍵OFF)
連續使用時間	8 小時
支援作業系統	藍芽：Android USB：系統Windows 7以上

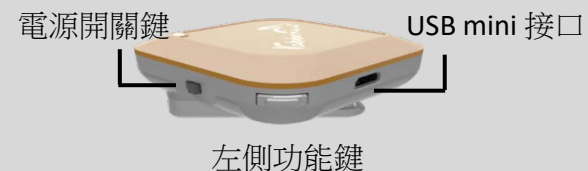
為了提高可靠性，還可以為每個軸配備更多的傳感器。一般而言IMU要安裝在被測物體的重心上。

。



# yabboni-操作功能介紹

電源開關鍵	單刀開關	On/off 標示
左側功能鍵	(短按1秒)	計數紀錄開始與結束(LED紅燈)
右側功能鍵	(短按1秒)	藍芽廣播開啟，與藍芽裝置配對(LED綠燈)
	(長按5秒)	電量顯示
LED電量指示燈號	(紅)	錄影指示燈、電量小於30%
	(橘)	關機指示燈、電量小於70%
	(綠)	配對指示燈、電量大於70%



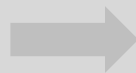
[綠燈閃爍]藍芽廣播中



[紅燈閃爍]計數記錄中



[長按右鍵5秒]可以確認電量狀態



電量大於70%



電量介於70% 到30%



電量小於30%



# gabboni-配件介紹

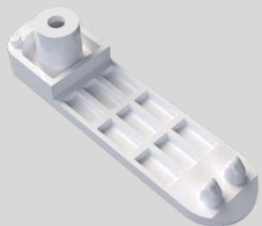


gabboni本體 (正面)



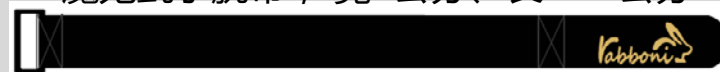
gabboni本體 (背面)

gabboni背夾(拆卸須將螺絲工具)



提供使用者跑步或行進間  
gabboni主體與鞋面穩固  
結合，確保動作的正確偵測。

魔鬼氈手腕帶，寬2公分、長27.5公分



提供使用者跑步或行進間gabboni主體  
與鞋面穩固結合，確保動作的正確偵測。

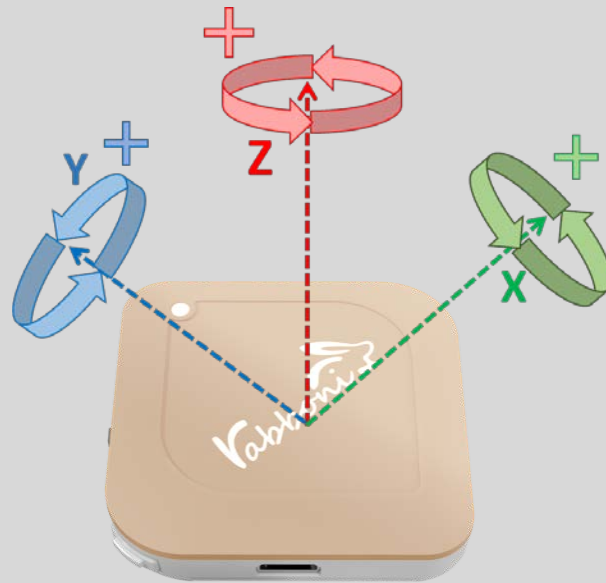
USB轉接線一條



USB Type A轉接 USB mini線，  
可提供傳輸數據以及充電功能。

# yabboni-軸向定義

直線軸：X/Y/Z 加速度 (Acceleration)  
環狀軸：X/Y/Z 角速度 (Gyro)





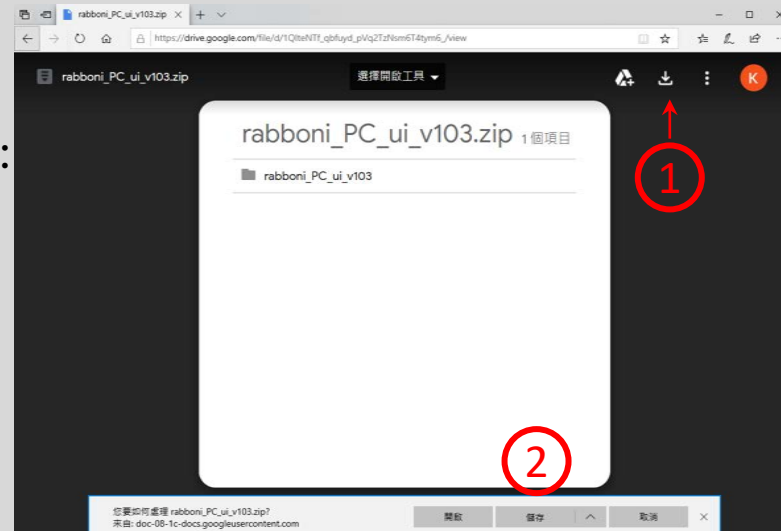
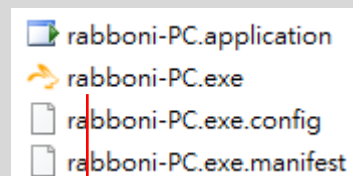
# yabboni PC UI 連線


1. rabboni\_pc\_UI下載/解壓縮資料夾(rabboni\_PC\_ui) :

<https://reurl.cc/QprO60>

2. 解壓縮檔中找到/建立捷徑

3. 執行 

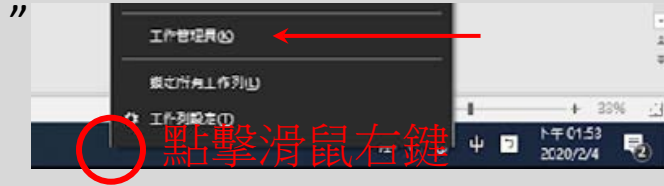


下載並解壓縮檔案  rabboni\_PC\_ui\_v103.zip

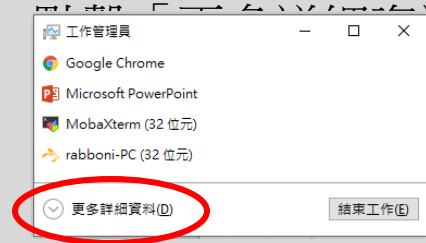


# 如果yabboni PC UI 連線程式無法開啟

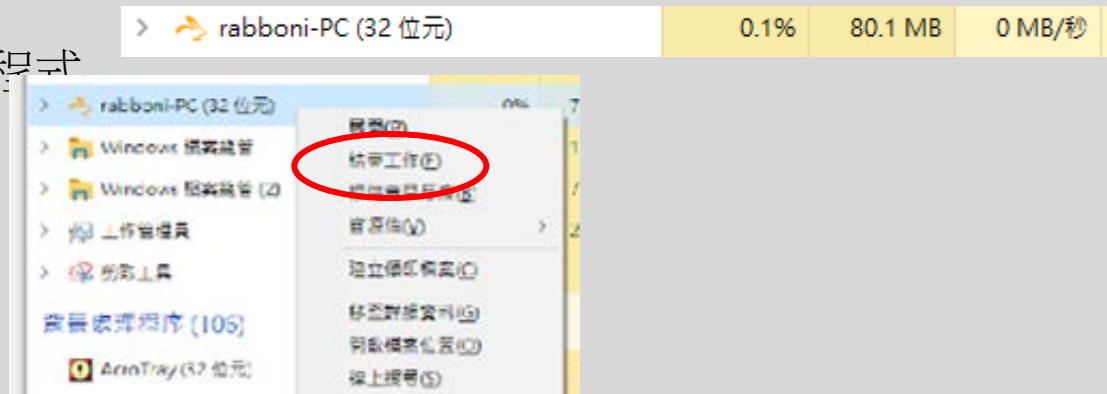
1. 執行工作管理員 (在工作列上按右鍵或同時按下Ctrl+Alt+Del，選擇”工作管理員



1. 工作管理員



1. 找到仍在背景執行的rabboni程式
2. 點擊右鍵選擇「結束工作」



# yabboni - PC UI介紹

The screenshot shows the yabboni PC UI with the following elements and callouts:

- 1:** USB connection button (checked).
- 2:** Bluetooth connection button (unchecked).
- 3:** MAC address input field (AB:CD:EF:GH:IJ).
- 4:** Scratch logo button.
- 5:** Scratch logo button.
- 6:** Device record count (0) with a Reset button.
- 7:** Drive status (Yes:1/No:0) with a value of 0.
- 8:** New record count (32) with a Reset button.
- 9:** Acceleration X, Y, and Z values: 0.031, -0.031, and 0.934.
- 10:** Angular velocity X, Y, and Z values: -1.251, 0.519, and 0.397.
- 11:** Parameter settings button.

1. USB: 點擊透過USB連線

2. Bluetooth: 點擊透過藍芽連線

3. MAC: 輸入裝置MAC的地方

4. Scratch: 點擊可以連到 Scratch

5. 驅動門檻: 設定內建加速度公式 $\sqrt{x^2 + y^2 + z^2}$ 並計算驅動次數結果的門檻(要大於多少算一次)

6. 裝置驅動記錄數/Reset: 紀錄驅動次數在

7. 驅動: 搖動超過門檻會回傳 1

8. 新驅動紀錄數/Reset: 每次重新連線回重新計數

9. X/Y/Z方向加速度 ( $1g=9.8m/sec^2$ )

10. X/Y/X方向角速度 (degree/sec)

11. 參數設定: 設定rabboni內的加速度以及角速度偵測範圍及 sampling rate。

# yabboni-USB連線

1. 打開Scratch UI
2. 連結USB



3. 點擊USB連結按鈕  
即可開始與電腦連線傳輸數據。

數字開始變動就是成功連線，變動數值就是三軸的加速度以及三軸的角速度。如果有問題的話就把檔案關起來重開。跳動值為量測值（含雜訊值），因此 Sensor 靜置仍會有跳動值。



Resource

# rabboni-配件介紹

1. 若電腦有開啟BLE 藍芽連線功能，會轉成藍色按鈕。(一般電腦筆電配備藍芽但不配備BLE 須加裝 BLE Dongle.)
2. 請輸入貼在盒子/裝置背後的MAC ID：AA:BB:CC:DD:EE:FF)
3. 點擊藍芽連線按鈕。

數字開始變動就是成功連線，變動數值就是三軸的加速度以及三軸的角速度。如果有問題的話就把檔案關起來重開。跳動值為量測值（含雜訊值），因此 Sensor 靜置仍會有跳動值。



1  
開啟BLE 藍芽連線



3



Resource

# yabboni-Scratch 連線

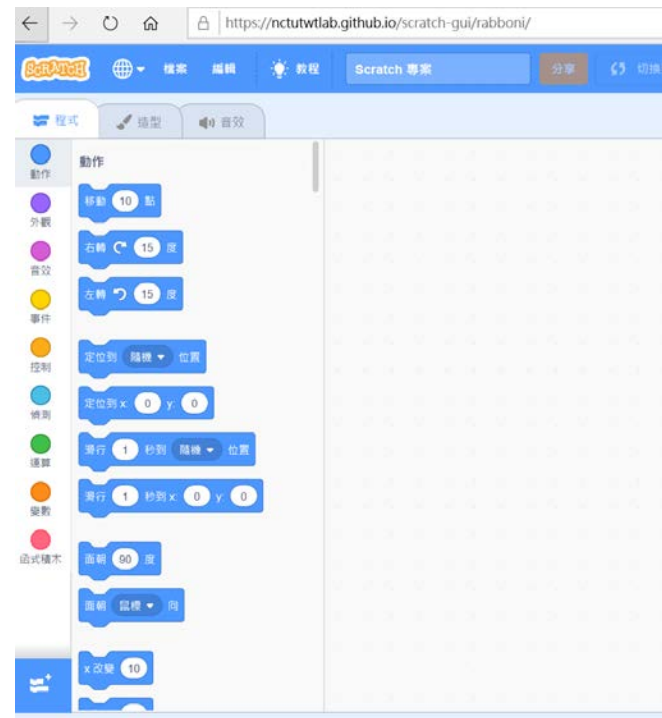
1. 點擊 Scratch Logo

或 2. 點擊連線

<https://nctutwlab.github.io/scratch-gui/rabboni/>

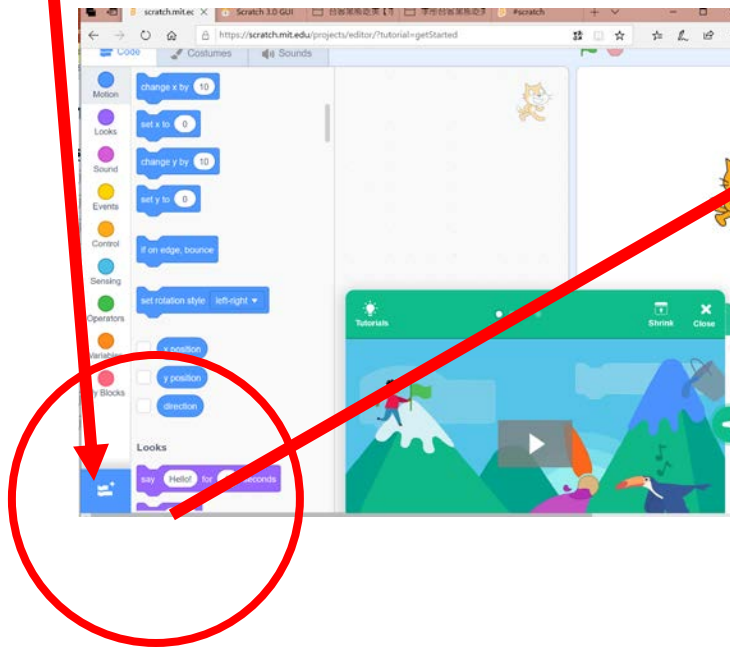


進入 Scratch 3.0

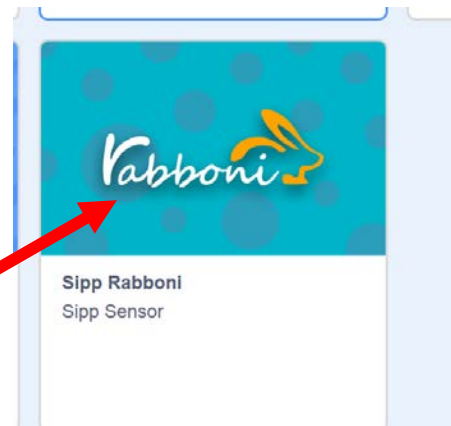


# rabboni 參數匯入

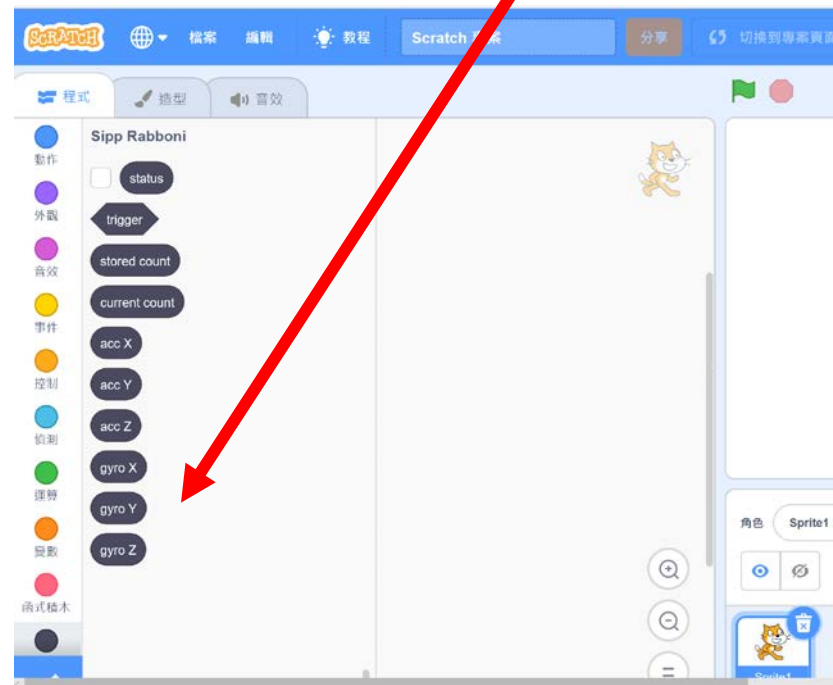
1. 點選 more Block



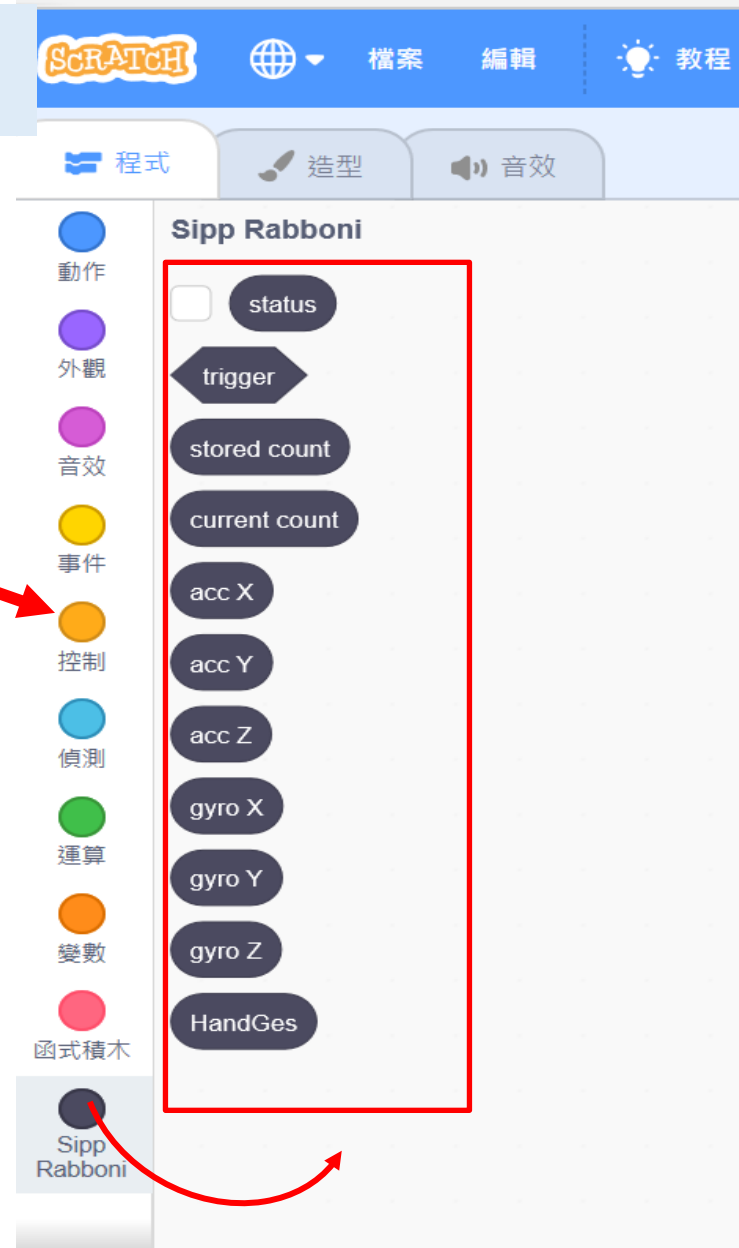
2 點選 Rabboni



3 參數匯入



# yabboni-Scratch 連線



點擊“ SIPP Rabboni ” 出現yabbonie感測值作為程式設計用

- StoredCount : 裝置記錄數
- Trigger : 驅動
- CurrentCount: 新紀錄數
- AccX : X方向加速度
- AccY : Y方向加速度
- AccZ : Z方向加速度
- GyroX : X方向角速度
- GyroY : Y方向角速度
- GyroZ : Z方向角速度

# 瘋狂 俄羅斯方塊

財金碩二 黃稜

AIOT coding, scratch3.0-rabboni





# [ 目錄 ]

CONTENTS

01 Demo 影片/設計概念

---

02 物件介紹

---

03 Code Introduction

---



A large, expressive green brushstroke that fills the left side of the slide. The stroke is composed of several overlapping, horizontal bands of varying shades of green, from a vibrant emerald to a darker, forest green. The texture is rough and painterly, with visible brush bristles and some white highlights where the stroke overlaps itself.

01



Demo 影片  
設計概念



# Demo 影片/設計概念

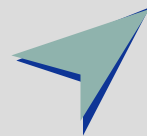
- 影片為經典遊戲俄羅斯方塊的 Demo 影片，改造成透過 **rabboni** 角速度 **x,z** 操作而下、左、右移動，變換方塊方向須透過 **rabboni trigger** 進行。
- 遊戲中可按下空白鍵暫停。
- 個人認為改造成 **rabboni** 的俄羅斯方塊敏感度較大，進行遊戲時須先適應一下。

---

<https://www.youtube.com/watch?v=O99axDFdsV8>

A large, expressive green brushstroke graphic that serves as a background for the number '02'. The stroke is thick and textured, with varying shades of green and some white highlights, giving it a hand-painted appearance. It is contained within a white, stylized frame that looks like a square with rounded corners and a slight shadow.

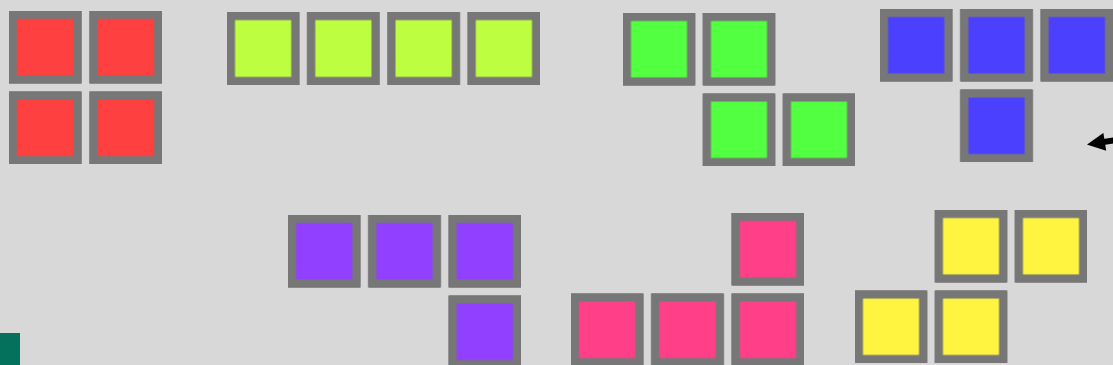
02



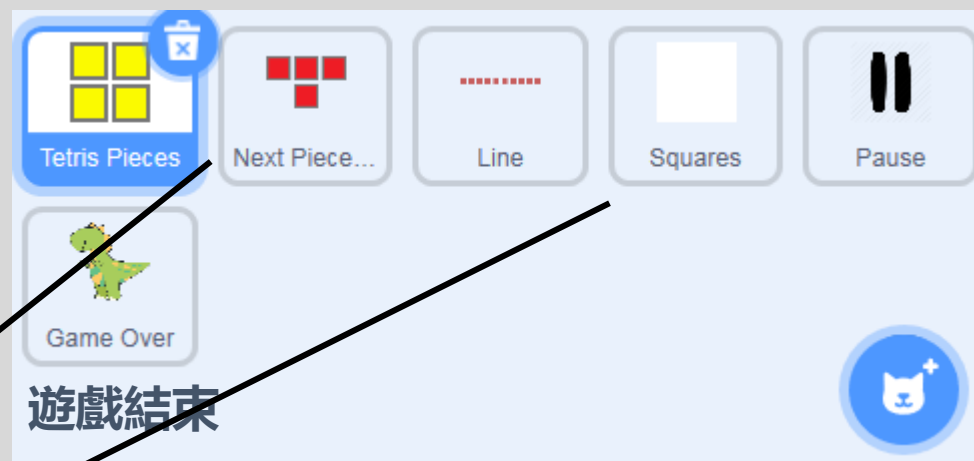
物件介紹

# 物件介紹

總共有7個物件



暫停鍵



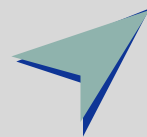
遊戲結束

顯示下一個會出現的物件(隨機出現)

消方塊標準



03



**Code Introduction**

# Code- 純掉落

## 設定不斷掉落的規則

當物件不斷碰到某些色塊 - 就會執行掉落

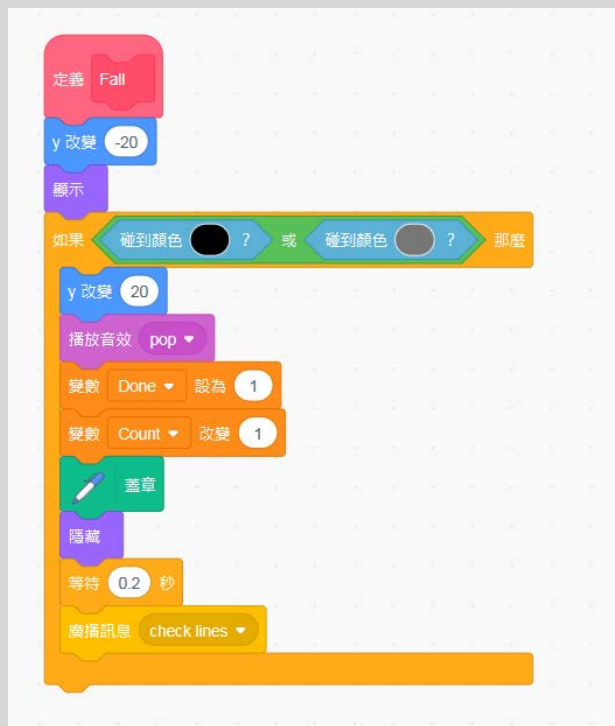
。

而當物件碰到頂端的色塊時，遊戲結束

## 預告下一次的方塊

在遊戲進行時，物件會隨機挑選下一個要出現的方塊，並且顯示在左上

由於設定遊戲時是從鍵盤開始設定，因此會有向下按鍵的設定，但不會影響遊戲進行。

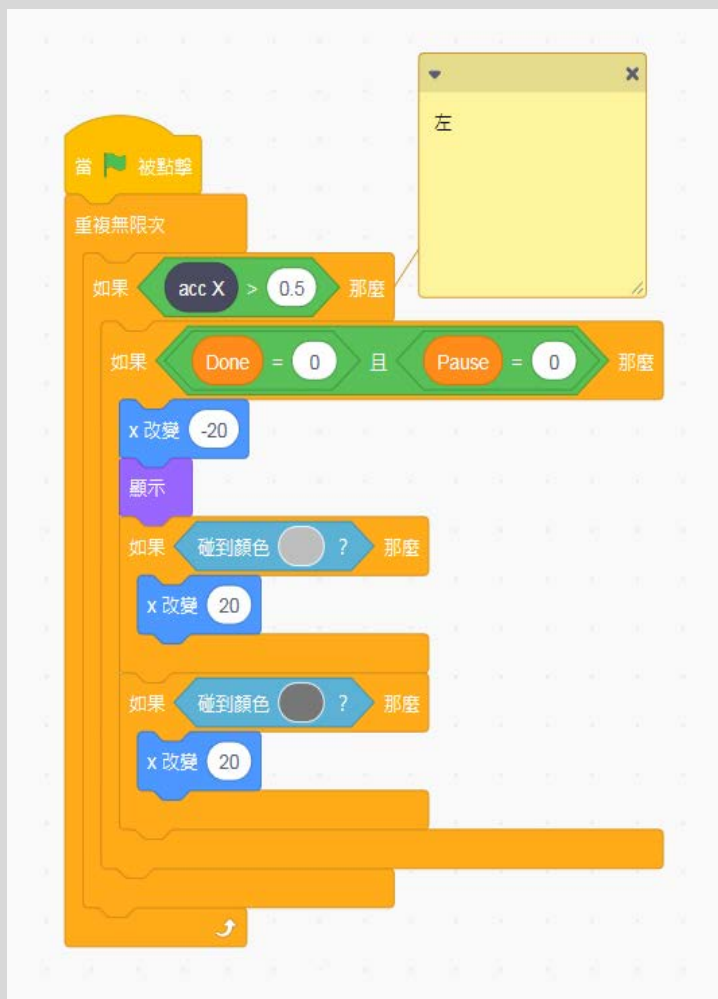


```
定義 Fall
y 改變 -20
顯示
如果 碰到顏色 黑色 ? 或 碰到顏色 灰色 ? 那麼
y 改變 20
播放音效 pop
變數 Done 設為 1
變數 Count 改變 1
蓋章
隱藏
等待 0.2 秒
廣播訊息 check lines
```



```
當收到訊息 start
定位到 x: 0 y: 180
造型換成 Next Piece Preview 的 造型編號
廣播訊息 switch
顯示
如果 碰到顏色 黑色 ? 那麼
廣播訊息 game over
停止 全部
變數 Done 設為 0
重複直到 Done = 1
如果 Pause = 0 那麼
如果 向下 鍵被按下? 不成立 那麼
Fall
等待 Time 秒
```

# Code- 方向

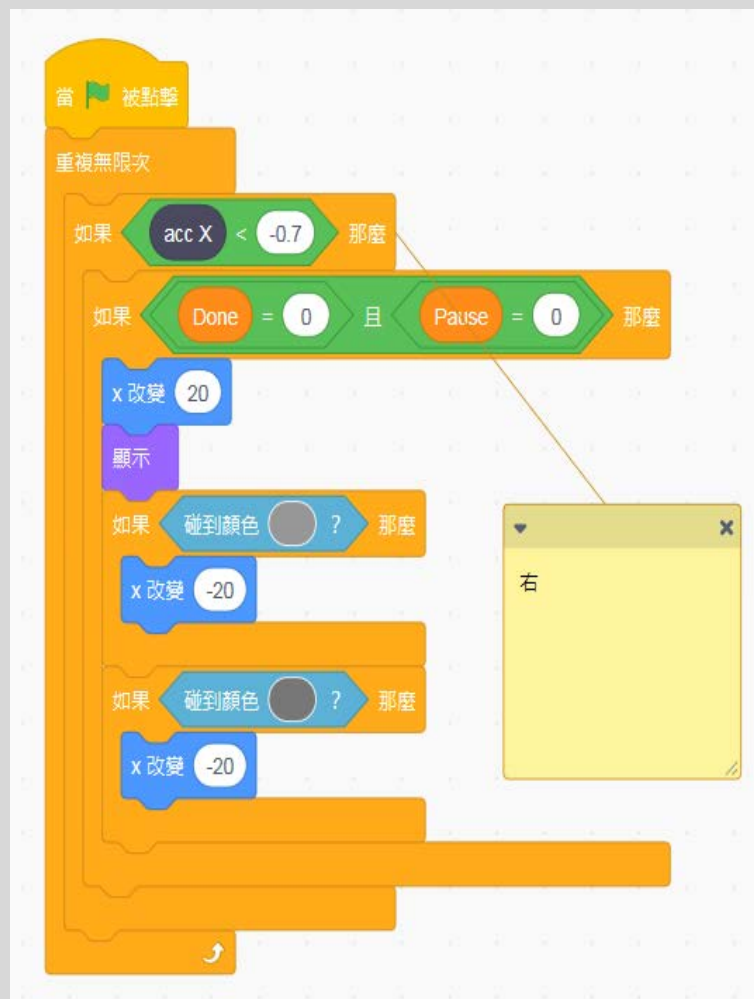


當 被點擊

重複無限次

- 如果  $\text{acc X} > 0.5$  那麼
- 如果  $\text{Done} = 0$  且  $\text{Pause} = 0$  那麼
  - x 改變 -20
  - 顯示
  - 如果 碰到顏色 ? 那麼
    - x 改變 20
  - 如果 碰到顏色 ? 那麼
    - x 改變 20

左

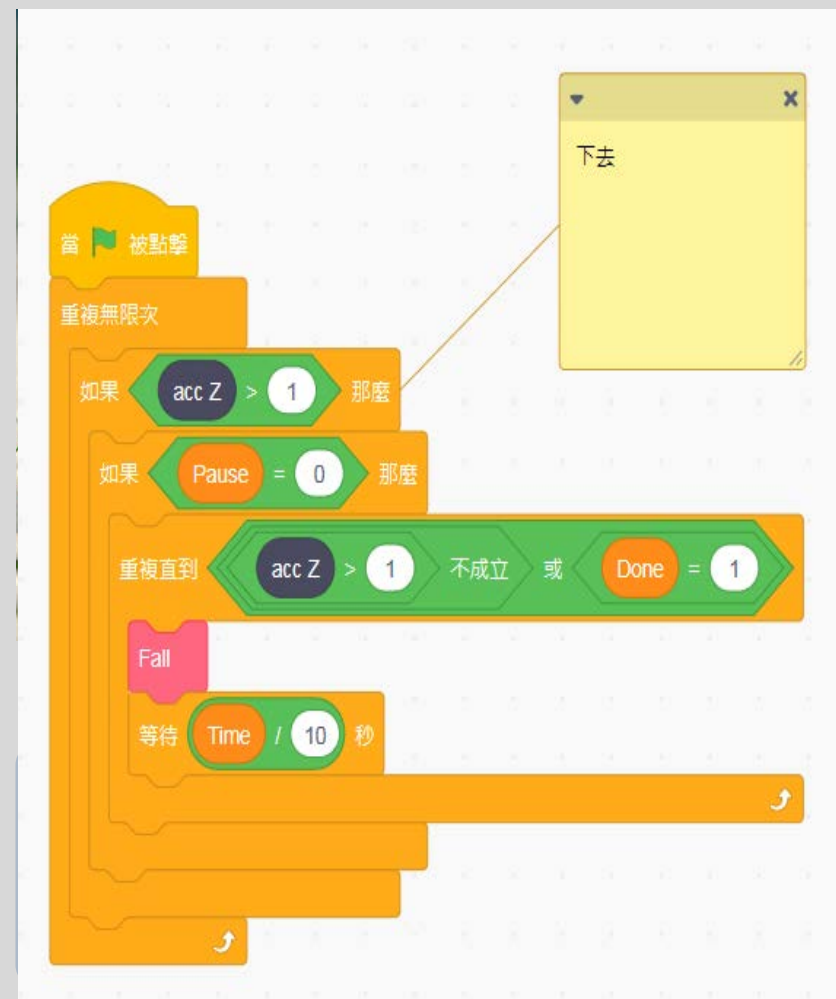


當 被點擊

重複無限次

- 如果  $\text{acc X} < -0.7$  那麼
- 如果  $\text{Done} = 0$  且  $\text{Pause} = 0$  那麼
  - x 改變 20
  - 顯示
  - 如果 碰到顏色 ? 那麼
    - x 改變 -20
  - 如果 碰到顏色 ? 那麼
    - x 改變 -20

右



當 被點擊

重複無限次

- 如果  $\text{acc Z} > 1$  那麼
- 如果  $\text{Pause} = 0$  那麼
- 重複直到  $\text{acc Z} > 1$  不成立 或  $\text{Done} = 1$
- Fall
- 等待  $\text{Time} / 10$  秒

下去



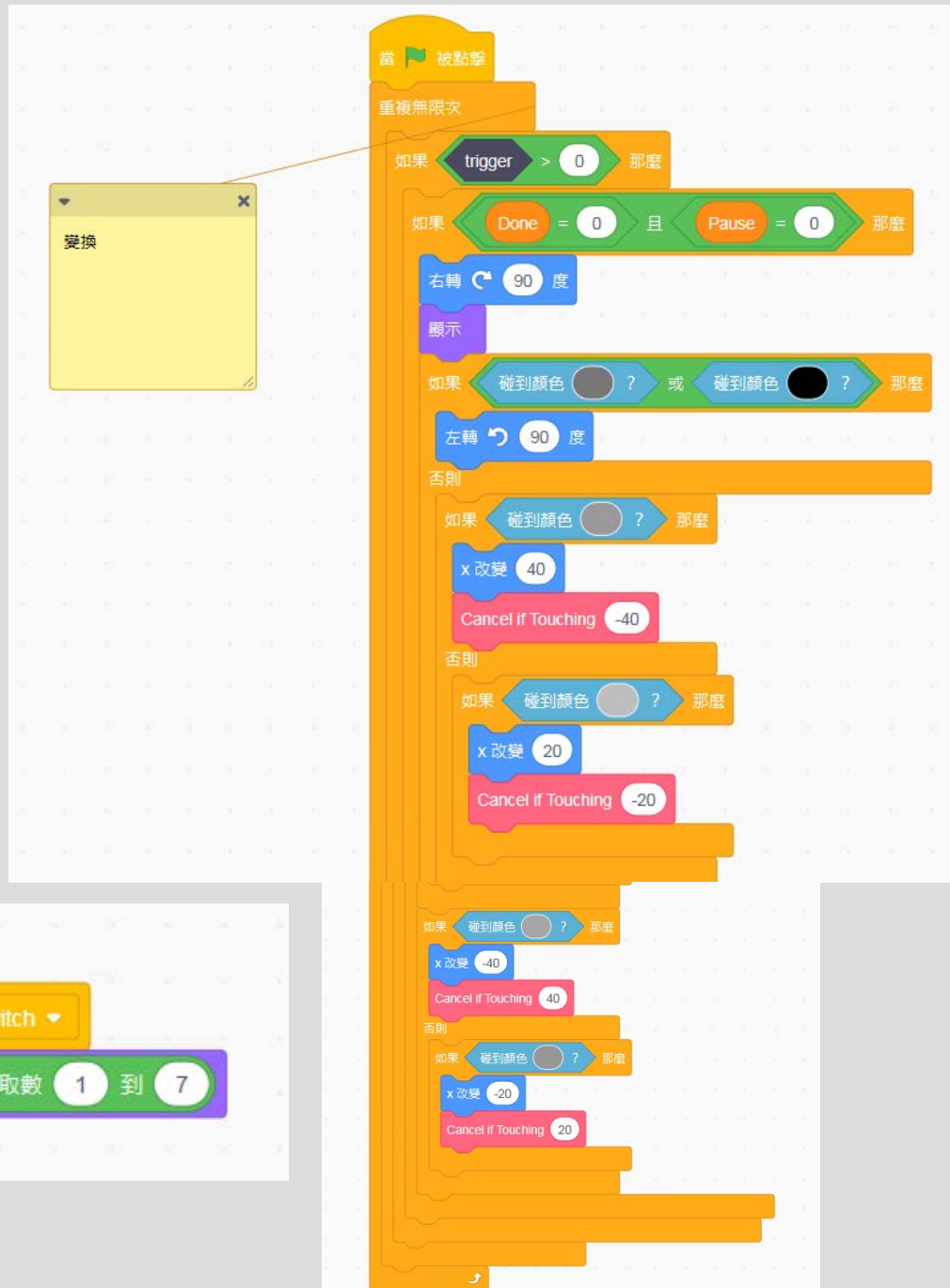
# Code- 轉換

## Trigger>0進行轉換

設計概念相似，當trigger>0方快就會進行90度轉換。

## 接下來出現的方塊顯示

如左下，switch的訊息出現，next物件就會隨機跳出。



# Code- 消方塊

當收到訊息 **check lines**

- 圖像效果 **ghost** 設為 **100**
- 變數 **Line Number** 設為 **0**
- y 設為 **-158**
- 重複 **17** 次
  - 顯示
  - 如果 碰到顏色 **○** ? 那麼
    - 變數 **Line Number** 改變 **1**
    - y 改變 **20**
  - 否則
    - 播放音效 **pop**
    - 變數 **Line Count** 改變 **1**
    - 廣播訊息 **clear line** 並等待
- 隱藏
- 廣播訊息 **start**

當 **被點擊**

- 隱藏
- 變數 **Line Count** 設為 **0**

當 **被點擊**

- 隱藏
- 當收到訊息 **clear line**
- 定位到 x: **90** y: **-150**
- y 改變 **Line Number \* 20**
- 顯示
- 重複 **17 - Line Number** 次
  - Copy Line**
  - y 改變 **20**
- 隱藏

定義 **Copy Line**

- 重複 **10** 次
  - y 改變 **20**
  - 顯示
  - 如果 碰到顏色 **●** ? 那麼
    - 造型換成 **red-0**
    - y 改變 **-20**
    - 蓋章
  - 否則
    - 如果 碰到顏色 **●** ? 那麼
      - 造型換成 **cyan-0**
      - y 改變 **-20**
      - 蓋章
    - 否則
      - 如果 碰到顏色 **●** ? 那麼
        - 造型換成 **blue-0**
        - y 改變 **-20**
        - 蓋章

# Code- 暫停與遊戲結束

## 隱藏線check 是否被完整填滿

續前頁，當隱藏線被填滿時，系統就會執行消去，右邊是增加消去時方塊色塊的動畫

## 暫停與遊戲結束

如右，按下空白鍵時會執行暫停，若物件碰到頂端會收到遊戲結束訊息。



A large, textured green brushstroke graphic in the background, consisting of several overlapping horizontal strokes of varying shades of green, creating a sense of movement and depth. The strokes are centered and extend across most of the width of the image.

**THANK YOU**