



Semi & AIOT Coding 智慧物聯-

以 Scratch 連結 **Yabboni** 介紹與操作

Date: 2025/4/30

Speaker: 電機14 王堃誠

Demo Video

備用影片網址



半導體
瑪利歐

Scratch

Rabboni

利用Scratch和Rabboni -進行瑪利歐小遊戲 :半導體瑪利歐

報告人: 王堃誠

電機14

Contents

γabboni-介紹

γabboni-感測參數介紹

γabboni-操作功能介紹

γabboni-配件介紹

γabboni-軸向定義

γabboni Scratch 連線

γabboni - Scratch UI介紹

γabboni-USB連線

γabboni-藍芽BLE 連線

γabboni-Scratch連線

γabboni-Scratch 範例程式



APPENDIX

γabboni-其他應用

<https://12u10.lab.nycu.edu.tw/>



IOT: Internet of Things



SENSORS *will be everywhere !!!!*

IMU:
Inertial Measurement Unit

- 加速度 (Accelerometer)
- 角速度 (Gyro)
- 磁力計 (Magnetometer)



半導體
Semiconductor

What is IMU ? Rabboni is an IMU.

Inertial Measurement Unit

慣性的

物體抗拒其運動狀態被改變的性質。

Accelerometer 加速規

測量移動 (加速度)

測量單位時間內速度變化

Gyroscope 陀螺儀

測量轉動 (角速度)

測量單位時間內角度變化

GeoMagnetic 地磁儀

測量地磁方向、大小

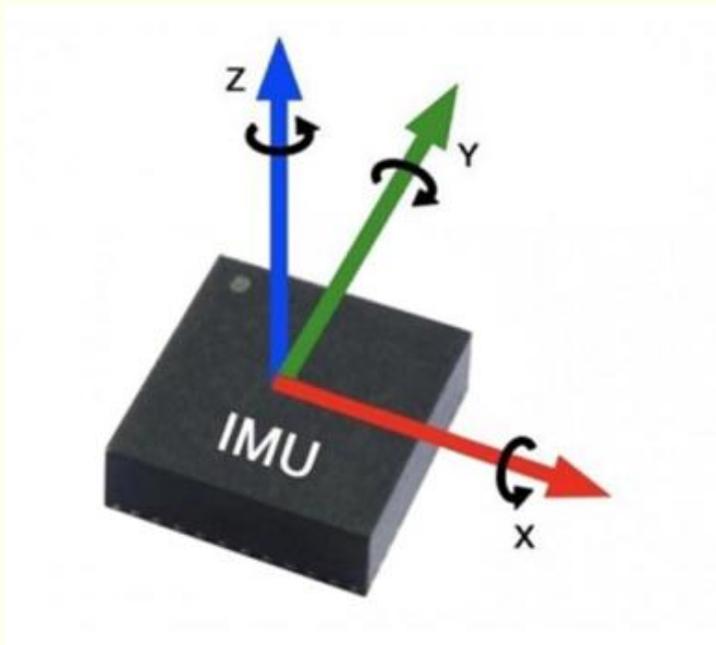
可用於定向



半導體
Semiconducto

r

Sensor 入門：聊聊半導體



聊聊 AIOT :
從 Rabboni + Scratch 開始 「貓兔同籠」



入門

資料親子



從 Rabboni  × SCRATCH 開始





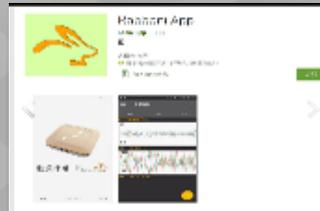
Rabboni is not just a device, It's a platform.

IMU 重力感測



內建六軸重力感測器

Data Extractor 重力感測數據擷取



Android
iOS

API 應用程式介面

- Scratch
- Python
- App Inventor
- Unity

AI Algorithm 演算法開發

- 行動偵測
- 姿態偵測
- 數據分析
- 訊號分析

AIoT 應用程式

- Sports
- Health
- Gaming
- Education



IoT 物聯/雲端介面

- Phone
- Pad
- NB
- Raspberry Pie

EDU 教育資源

- 企業社會責任
- 大學社會責任
- 縣市教育局處合作

r



- yabboni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
- 配有LED燈，指示rabboni運作狀態及電量顯示



介紹

電池容量 充電方式	120mAh 鋰離子充電電池 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



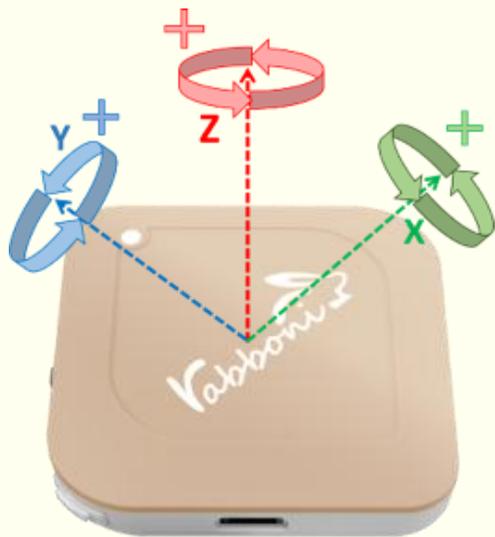
直接看我操作

!



感測參數及軸向介紹

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



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



yabboni 配件介紹



yabboni本體 (正面)



yabboni本體 (背面)



USB轉接線一條



yabboni背夾



魔鬼氈手腕帶 (寬2公分、長27.5公分)

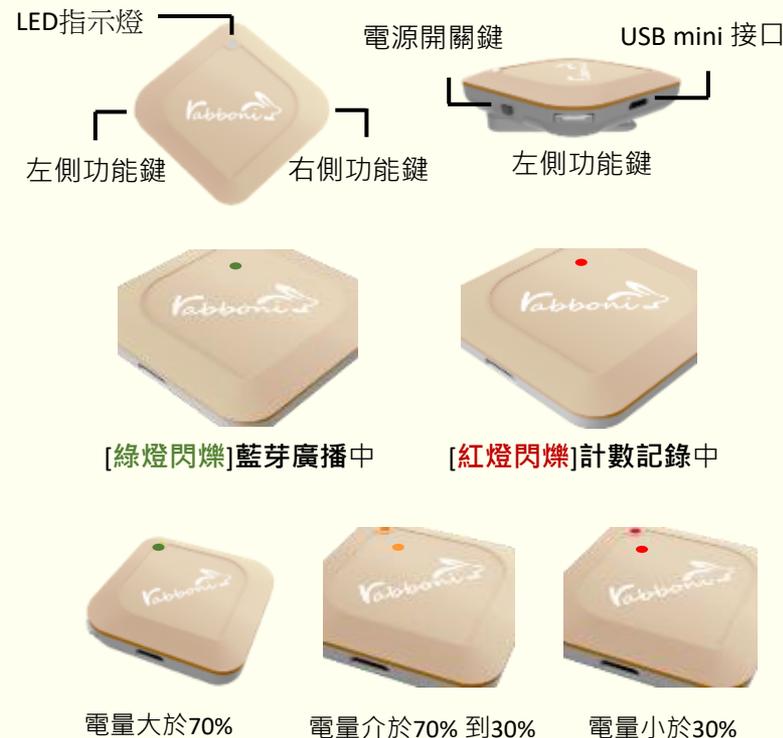
- USB Type A轉接 USB mini線，可提供傳輸數據以及充電功能。

- 提供使用者跑步或行進間yabboni主體與鞋面穩固結合，確保動作正確偵測



Kabboni 配件介紹

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%



[長按右鍵5秒]可以確認電量狀態





安裝安裝.....再安裝



Scratch桌面板多連使用說明



半導體

Semiconducto

r

Scratch 桌面板多連程式下載

1. 進入連結：<https://12u10.lab.nycu.edu.tw/downloads/>
2. 如果出現警告，選擇“仍要下載”
3. 選擇“儲存”



下載專區

一般檔案下載

教材內容	
<p>Scratch</p> <p>Scratch 是麻省理工媒體實驗室終身幼稚園組開發的一套電腦程式開發平台，旨在讓程式設計語言初學者不需先學習語言法便能設計產品。</p>	檢視+
<p>Python</p> <p>Python 是種廣泛使用的直譯式、進階程式、通用型程式語言，有吉多·范羅蘇姆創造，第一版釋出於1991年。</p>	檢視+ 1
<p>Rabonni AIoT 程式教育感測裝置申請表</p>	檢視+
<p>Rabboni API</p> <p>文文孟參考資料 API 3.0 含有Scratch,Python,APInventor,Raspberry Pi, Arduino等套件</p>	檢視+

<p>Scratch</p> <p>Scratch 是麻省理工媒體實驗室終身幼稚園組開發的一套電腦程式開發平台，旨在讓程式設計語言初學者不需先學習語言法便能設計產品。</p>	檢視-
<p>教材名稱：Rabboni-scratch 作者：溫環岸 簡章介紹：本教材為scratch連結rabboni介紹與操作教學。</p>	下載 次數：1333
<p>教材名稱：Rabboni-scratch + Sensor 作者：溫環岸 簡章介紹：本教材為scratch連結rabboni介紹與操作教學，並結合感測參數及製程介紹。</p>	下載 次數：938
<p>教材名稱：10/26 師培Rabboni-scratch + Sensor 作者：朱保銘 簡章介紹：本教材為scratch一對多連結rabboni介紹與操作教學。</p>	下載 次數：262
<p>教材名稱：10/28 師培Rabboni-scratch + Sensor 作者：林志威 簡章介紹：本教材為scratch一對多連結rabboni介紹與操作教學。</p>	下載 次數：247
<p>教材名稱：11/02 師培Rabboni-scratch + Sensor 作者：游天維 簡章介紹：本教材為scratch一對多連結rabboni介紹與操作教學。</p>	下載 次數：366
<p>教材名稱：AIOT Coding 智慧物聯師培分章(中小學) 作者：溫環岸 簡章介紹：本教材為scratch連結rabboni介紹與操作教學，並結合感測參數及製程介紹。</p>	下載 2 → 連結 次數：37



SCRATCH 桌面板多連程式下載

1. 進入連結：<https://12u10.lab.nycu.edu.tw/downloads/>
2. 如果出現警告，選擇“仍要下載”
3. 選擇“儲存”



2. 如果出現警告，選擇“仍要下載”





Scratch桌面板多連使用說明-USB連線



SCRATCH 桌面板多連使用說明-USB連線

STEP 1

解壓縮後，打開資料夾，點擊應用程式開啟



STEP 2

開啟應用程式



點擊新增裝置



!!! 注意 !!!

USB只能連線1個rabboni

藍芽最多同時4個 rabboni

STEP 3

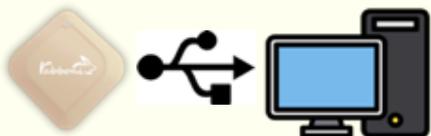
出現選擇連線方式視窗，
可以選擇USB或藍芽連線



SCRATCH 桌面板多連使用說明-USB連線

STEP 4

連結USB與電腦



STEP 5

點擊USB的選項



STEP 6

選擇裝置



STEP 7

選擇 Rabboni – USB HID UART Bridge



STEP 8

選擇「確認」



SCRATCH 桌面板多連使用說明-USB連線

STEP 9

數字跳動代表連線成功



STEP 10

可以編輯裝置在電腦上的名稱，
名稱會對應到Scratch裡



STEP 11

按下「確認」後，名稱改變





Scratch桌面板多連使用說明-藍芽連線



SCRATCH 桌面板多連使用說明-藍芽連線

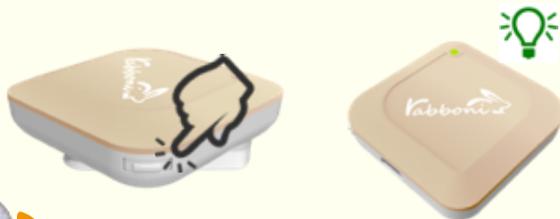
STEP 1

- 電腦沒有藍芽:連結dongle與電腦
- 電腦沒有藍芽:請確認藍芽在4.0-5.1間



STEP 2

短按**右鍵**1秒，開始藍芽連線，綠燈會閃爍直到配對成功。若無配對到手機，會自動於30秒後停止廣播。



藍芽連線手機成功後，**綠燈**每10秒閃爍一次

半導體
Semiconductor

STEP 3

點擊「藍芽」的選項



STEP 4

選擇裝置



STEP 5

選擇欲連結rabboni裝置的MAC碼



MAC碼在rabboni的本體背面

STEP 6

選擇「確認」



SCRATCH 桌面板多連使用說明-藍芽連線

STEP 7

數字跳動代表連線成功



STEP 8

可以編輯裝置在電腦上的名稱，
名稱會對應到Scartch裡



STEP 9

按下「確認」後，名稱改變



SCRATCH 桌面板多連使用說明-藍芽連線

STEP 10

新增其他裝置



點擊新增更多裝置



SCRATCH 桌面板多連使用說明-藍芽連線

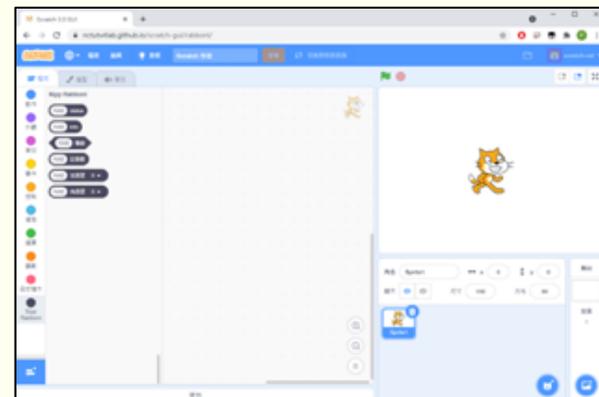
STEP 10

點擊左邊Scratch的ICON



STEP 11

點擊Scratch的ICON，跳轉到瀏覽器



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

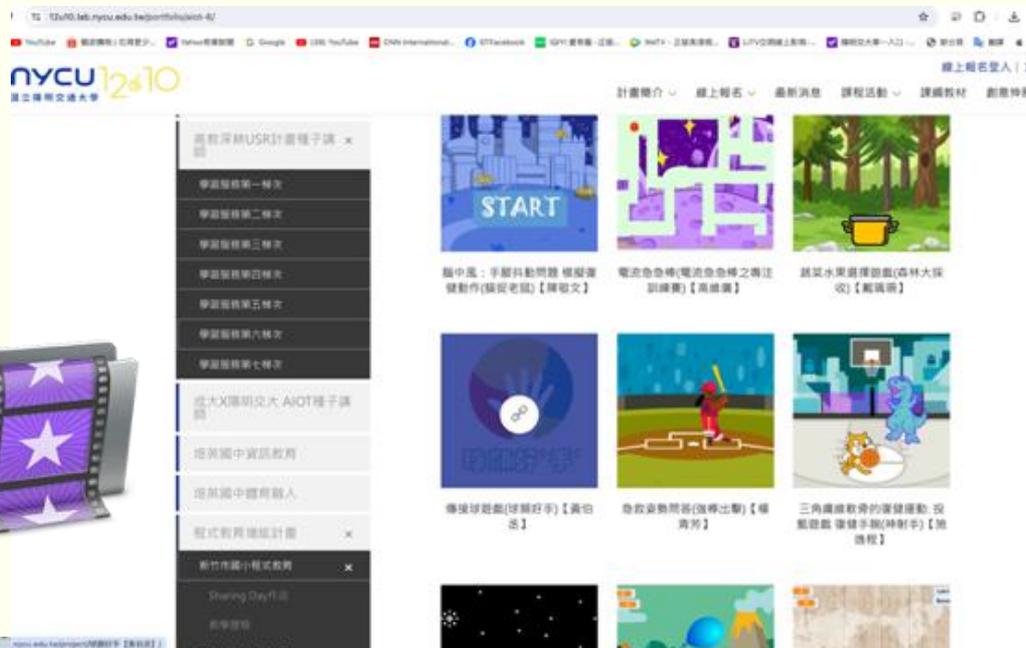


半導體
Semiconductor

程式觀摩:

從別人設計的程式開始: 到『12u10 一定要你贏』網站去逛逛喔

<https://12u10.lab.nycu.edu.tw/portfolio/aiot-6/>



創意奔放
SHOW Time!!



Scratch

半導體
瑪利歐

Rabboni

利用Scratch和Rabboni -進行瑪利歐小遊戲 :半導體瑪利歐

報告人: 王堃誠

電機14



目錄 / CONTENTS



發想



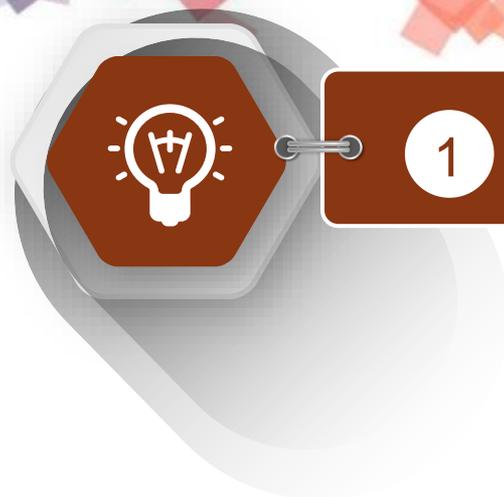
遊戲簡介



遊戲影片

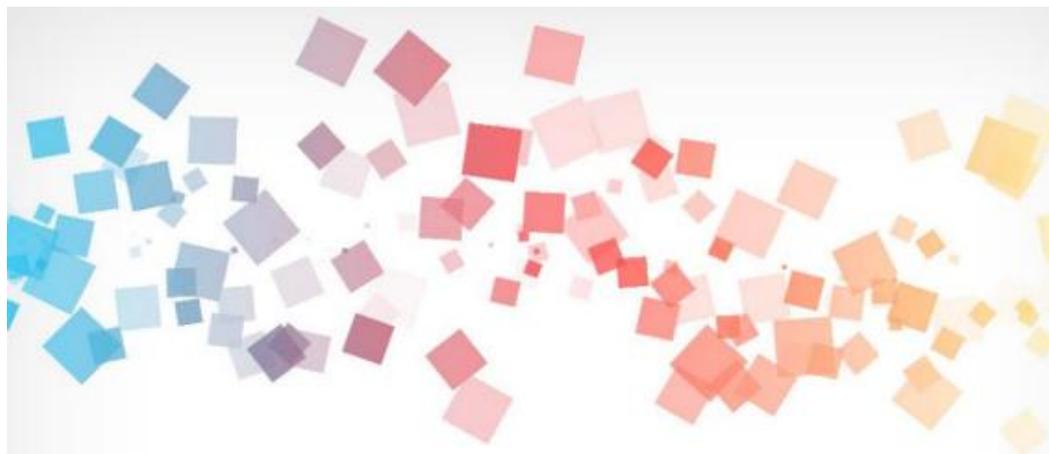


程式介紹



1

發想



發想

這款遊戲的設計靈感來自半導體的基礎原理——摻雜的應用。
Si若想成為半導體，就得透過P型或者N行摻雜，其中還細分成輕度
重度摻雜等等。

由於半導體是整個現代晶片的基礎，我希望通過這款遊戲，讓玩家
更直觀地理解半導體摻雜的過程。



》》》製作原因

這款遊戲旨在結合教育性和互動性，利用Rabboni 傳感器讓玩家在操作過程中直觀地理解傳感器的運作方式，同時了解到半導體摻雜的特性，達到動手操作與半導體學習的雙重效果。





》》》遊戲玩法簡介

遊戲玩法簡介

瑪莉歐小遊戲，不過一切都變得更半導體相關。金幣變成了電子，蘑菇變成了磷原子，代表了輕度P型摻雜，而火焰花則變成了砷原子，代表重度P型摻雜。Goomba變成了絕緣體，烏龜變成了導體。玩家要操作Si瑪莉歐，一步一步邁向終點!

操控說明

Walk/Jump - Arrow Keys, WASD or Tilt Rabboni

Throw Electron(Arsenic doping) - Space or Shake Rabboni





遊戲影片

備用影片網址





▶▶▶ 程式介紹-主要角色

The screenshot shows a game engine interface. The top part displays a game scene with a blue sky, green hills, and a Mario character. Text at the top reads: "Thanks for playing!!! Again, this game is remix of Mario by Will_Wam, adapted by Kun Cheng". Below the scene are several icons: a blue 'W' in a square, a black 'X', a character icon, and a Mario icon. The bottom part of the interface is a control panel with the following elements:

- 角色 (Character):** Mario
- 位置 (Position):** x: 0, y: -110
- 顯示 (Display):** On (eye icon), Visible (lock icon)
- 尺寸 (Size):** 100
- 方向 (Direction):** 90
- 舞台 (Stage):** 1
- 背景 (Background):** 1
- 物件列表 (Object List):** Mario, All Tiles, Pen rende..., Other Clon..., Intro

Mario

主角色

ALL Tiles

遊戲中的不可移動 Instances

Pen rendereerrrr

遮住渲染中的最左邊以及最右邊的 Instances

Other Clones

遊戲中移動的 Instances

Intro

負責Display遊戲的結束畫面

背景-程式碼(僅介紹有使用到的)

控制遊戲背景BGM

The image shows a collection of Scratch code blocks organized into several groups. The blocks are as follows:

- Group 1:** "當收到訊息 ya just died boi" (When I receive the message 'ya just died boi'), "音量設為 0 %" (Set volume to 0%), "停播所有音效" (Stop all sounds).
- Group 2:** "當收到訊息 restart level" (When I receive the message 'restart level'), "停播所有音效" (Stop all sounds), "音量設為 75 %" (Set volume to 75%), "重複無限次" (Repeat forever) loop containing "播放音效 smb_main-theme 直到結束" (Play sound effect 'smb_main-theme' until finished).
- Group 3:** "當收到訊息 start" (When I receive the message 'start'), "音量設為 75 %" (Set volume to 75%), "廣播訊息 restart level" (Broadcast message 'restart level').
- Group 4:** "當收到訊息 win!" (When I receive the message 'win!'), "音量設為 75 %" (Set volume to 75%), "播放音效 smb_stage_clear 直到結束" (Play sound effect 'smb_stage_clear' until finished), "廣播訊息 make some fireworks 並等待" (Broadcast message 'make some fireworks' and wait), "等待 1.5 秒" (Wait 1.5 seconds), "廣播訊息 darkness" (Broadcast message 'darkness').
- Group 5:** "當收到訊息 hop on flag pole! d" (When I receive the message 'hop on flag pole! d'), "停止 other scripts in stage" (Stop other scripts in stage), "停播所有音效" (Stop all sounds), "重複無限次" (Repeat forever) loop containing "計時器重置" (Reset timer).
- Group 6:** "當收到訊息 restart level" (When I receive the message 'restart level'), "重複無限次" (Repeat forever) loop containing: "變數 Mario Still able to break block on this frame 設為 1" (Set variable 'Mario Still able to break block on this frame' to 1), "等待直到 Mario Still able to break block on this frame = 1 不成立" (Wait until 'Mario Still able to break block on this frame' = 1 is no longer true), "等待 0.05 秒" (Wait 0.05 seconds).
- Group 7:** "當收到訊息 start" (When I receive the message 'start'), "變數 Show Hitboxes 設為 0" (Set variable 'Show Hitboxes' to 0), "變數 Edge 設為 1" (Set variable 'Edge' to 1).
- Group 8:** "當收到訊息 restart level" (When I receive the message 'restart level'), "重複無限次" (Repeat forever) loop containing: "變數 Still able to spawn enemy? 設為 1" (Set variable 'Still able to spawn enemy?' to 1), "計時器重置" (Reset timer).

控制部分全局變數(ex: 敵人是否能生成)

Mario-程式碼

遊戲開始以及重生的無敵時間

當發生bug卡出地圖外時，傳送回安全地帶

地形碰撞檢測，以便觸發跳躍停下、側碰反彈等行為。

The code blocks are as follows:

- Game Start/Respawn:**
 - 當收到訊息 restart level: 等待 0.2 秒, 變數 CANNOT DIE 設為 0.
 - 當 被點擊: 變數 CANNOT DIE 設為 0.
- Game Initialization (official start!):**
 - 變數 MARIO'S TYPE 設為 1, Vertical HitBox 設為 13, Horizontal HitBox 設為 8, Movements Frozen 設為 0, 音量設為 60%, 變數 SCROLL X 設為 0.
 - x 設為 0, y 設為 -110.
 - 顯示, 圖層移到 最上 層, 面朝 90 度.
 - 變數 Temp Invincible 設為 0, # of fireballs out 設為 0, 變數 Xv 設為 0, 變數 Yv 設為 0, x 改變 1.
- Collision Detection:**
 - 當 timer > 1: 隱藏, 定義 fdfsdfsfsa, 重複直到 TOUCHING GROUND = 1, 定位到 x: 隨機取數 (-240 到 240) y: 隨機取數 (-110 到 180), Touching ground VERTICALLY?.
 - 定義 SENSE: 變數 scroll# 設為 1, 變數 U IS TOUCHIN SOMETHIN 設為 0, 重複 清單 >> DIR 的長度 次, 如果 >> TYPE 的 第 scroll# 項 = 2 或 >> TYPE 的 第 scroll# 項 = 100 或 >> TYPE 的 第 scroll# 項 > 90 或不成立 那麼, 如果 >> X POS 的 第 scroll# 項 + >> RIGHT HB 的 第 scroll# 項 > x 座標 - Horizontal HitBox 且 >> X POS 的 第 scroll# 項 + >> LEFT HB, 變數 U IS TOUCHIN SOMETHIN 設為 1, 變數 scroll# 改變 1.

遊戲正式開始時，Mario 設定成「小 Mario」、初始化他的碰撞盒大小、解除凍結、設好音量、滾動參數 (scroll X)、初始位置、圖層與朝向，並重置臨時無敵、火球數、速度等所有關鍵變數，最後用 x 改變 1 開始持續往右捲動畫面。

➤➤➤ Mario-程式碼

延長造型切換間隔

大小改變時變更HitBox大小

Debug用·顯示HitBox範圍

The image shows a collection of Scratch code blocks for a Mario game. The blocks are organized into three sections:

- Top Section (Movement and State):** Includes a 'when green flag clicked' block, a 'wait until' block for right or left key presses, a 'Unit Change Costume' block with a value of 1000, and a 'wait until' block for key presses or 'not true'.
- Middle Section (HitBox Adjustment):** Contains three 'when received message' blocks:
 - 'u grow big and strong yuussu': Sets 'Vertical HitBox' to 23, changes 'Vertical HitBox' to 3, and sets 'Vertical HitBox' to 23.
 - 'u get small x': Sets 'MARIO'S TYPE' to 1, sets 'Vertical HitBox' to 13, changes 'Vertical HitBox' to -3, and sets 'Vertical HitBox' to 13.
- Bottom Section (Debug and Mushroom Logic):** Includes a 'when green flag clicked' block with a 'show' block, a 'hitbox' block, and a 'display hitbox' block. The 'display hitbox' block shows 'Mario' with a 'y' coordinate of 224 and an 'x' coordinate of 32. It also includes a 'when received message' block for 'u grow big and strong yuussu' with various effects like 'Movements Frozen', 'size', 'jump power', 'MARIO'S TYPE', and 'costume' changes.

當發生bug卡出地圖外時，傳送回安全地帶

吃到蘑菇(磷原子)時的變大動畫

Mario-程式碼

This Scratch code block defines the movement and costume logic for Mario. It starts with a 'Costume Stuff' block. The logic is as follows:

- Initial state: `Movements Frozen = 0`.
- Initial velocity: `y 改變 Yv = 5`.
- SENSE** block: `UI IS TOUCHIN SOMETHIN = 1`.
- IF** block: `向下 鍵被按下? 或 s 鍵被按下? 或 RAB 加速度 Y = -0.7 且 MARIO'S TYPE = 1 不成立`.
 - True: `MARIO'S TYPE = 2`, `變數 Costume = 設為 16`.
 - False: `MARIO'S TYPE = 3`, `變數 Costume = 設為 26`.
- IF** block: `向右 鍵被按下? 或 d 鍵被按下? 或 RAB 加速度 X = -0.7 且 向左 鍵被按下? 或 a 鍵被按下? 不成立`.
 - True: `旋轉 90 度`.
 - False: `如果 Xv < 0`.
 - True: `如果 Costume = 17 * MARIO'S TYPE 不成立`.
 - True: `變數 Costume = 設為 17 * MARIO'S TYPE`, `播放音效 Skid`.
 - False: `rest of stuff`.
- IF** block: `向左 鍵被按下? 或 a 鍵被按下? 或 RAB 加速度 X = 0.7 且 向右 鍵被按下? 或 d 鍵被按下? 不成立`.
 - True: `旋轉 -90 度`.
 - False: `如果 Xv > 0`.
 - True: `如果 Costume = 17 * MARIO'S TYPE 不成立`.
 - True: `變數 Costume = 設為 17 * MARIO'S TYPE`, `播放音效 Skid`.
 - False: `rest of stuff`.
- Final velocity: `y 改變 Yv = -1 * 5`.

This Scratch code block defines the movement and costume logic for Rabboni. The logic is as follows:

- IF** block: `向左 鍵被按下? 或 a 鍵被按下? 或 RAB 加速度 X = 0.7 且 向右 鍵被按下? 或 d 鍵被按下? 不成立`.
 - True: `旋轉 -90 度`.
 - False: `如果 Xv > 0`.
 - True: `如果 Costume = 17 * MARIO'S TYPE 不成立`.
 - True: `變數 Costume = 設為 17 * MARIO'S TYPE`, `播放音效 Skid`.
 - False: `rest of stuff`.
- IF** block: `向右 鍵被按下? 或 d 鍵被按下? 或 RAB 加速度 X = -0.7 且 向左 鍵被按下? 或 a 鍵被按下? 不成立`.
 - True: `旋轉 90 度`.
 - False: `如果 MARIO'S TYPE = 1`.
 - True: `變數 Costume = 設為 6`.
 - False: `如果 MARIO'S TYPE = 2`.
 - True: `變數 Costume = 設為 15`.
 - False: `如果 MARIO'S TYPE = 3`.
 - True: `變數 Costume = 設為 25`.
 - Final velocity: `y 改變 Yv = -1 * 5`.

上下左右/WSAD/操控Rabboni 時的造型變換

Mario-程式碼

This Scratch code block contains the logic for Mario's movement and jumping. It starts with a scroll speed of 0 and a jump height of 1. It checks for movement freezing and updates vertical velocity (Yv) based on ground contact. A sense block checks for movement directions (up, down, left, right) and updates Mario's speed. It also checks for the RAB (Rabbit) power-up, which increases Mario's speed and jump power.

This Scratch code block handles Mario's jump and touch events. It checks if Mario is touching something (ground or a block). If touching ground, it resets vertical velocity and checks for movement directions. If touching a block, it checks if the block is above Mario and if he is jumping. It also checks for the RAB power-up and updates jump power. A sense block checks for the RAB power-up and updates Mario's speed and jump power.

上下左右/WSAD/操控Rabboni 時的角色動作控制(移動/跳躍)

▶▶▶ Mario-程式碼

死亡動畫，切換造型

```

當收到訊息 ya just died boi
造型換成 Die
變數 Movements Frozen 設為 1
變數 Xv 設為 0
變數 Yv 設為 12
等待 0.5 秒
重複直到 y 座標 < -165
  變數 Yv 改變 -0.8
  y 改變 Yv
  播放
  
```

獲得火焰花(As摻雜)時的
動畫

```

當收到訊息 fire flower gotten >.d
播放音效 smb_powerup
變數 MARIO'S TYPE 設為 3
變數 Costume 設為 21
重複直到 Vertical HitBox > 23
  變數 Vertical HitBox 改變 3
  變數 Vertical HitBox 設為 23
  
```

大瑪莉歐受傷時縮小回小瑪莉歐

```

當收到訊息 u got small (
變數 Movements Frozen 設為 1
尺寸設為 100 %
變數 MARIO'S TYPE 設為 1
y 改變 -10
重複 4 次
  造型換成 big while growing
  y 改變 0
  圖像效果 brightness 設為 0
  等待 AAA will delete 秒
  造型換成 Stand
  y 改變 0
  圖像效果 brightness 設為 25
  等待 AAA will delete 秒
  圖像效果清除
  變數 Costume 設為 2
  變數 Movements Frozen 設為 0
  
```

```

當收到訊息 get hit
如果 CANNOT DIE = 0 那麼
  變數 Movements Frozen 設為 1
  變數 Xv 設為 0
  變數 Temp Invincible 設為 1
  如果 MARIO'S TYPE = 1 或 y 座標 < -160 那麼
    變數 CANNOT DIE 設為 1
    廣播訊息 ya just died boi
    等待 0.04 秒
    播放音效 smb_mariodies 直到結束
    等待 0.5 秒
    廣播訊息 restart level
    等待 0.2 秒
    變數 Temp Invincible 設為 0
    變數 CANNOT DIE 設為 0
  否則
    播放音效 Warp
    廣播訊息 u got small ( 並等待
    變數 Movements Frozen 設為 0
    等待 0.6 秒
    變數 Temp Invincible 設為 0
  
```

暫時無敵 + 凍結移動：避免一次受傷造成重複觸發。
小 Mario 或空中跌落時 → 設為不可受傷，
死亡後重啟關卡；
small：觸發縮小流程，之後解除凍結和無敵狀態。

▶▶▶ Mario-程式碼

```

當收到訊息 hop on flag pole! :d
變數 Movements Frozen 設為 1
變數 SCROLL X 設為 -188.5846
等待 0.05 秒
x 設為 -5
播放音效 smb_flagpole
如果 MARIO'S TYPE = 1 那麼
    面朝 90 度
    造型換成 Flagside
    變數 Movements Frozen 設為 1
    重複直到 y 座標 < -80
        y 改變 -5
    等待 0.5 秒
    面朝 -90 度
    x 改變 20
    等待 0.5 秒
    面朝 90 度
    造型換成 Big Jump
    變數 Xv 設為 4
    變數 Yv 設為 12
    重複直到 y 座標 = -100
        變數 Yv 改變 -1.7
        y 改變 Yv
        x 改變 Xv
        如果 y 座標 < -100 那麼
            y 設為 -100
    造型換成 Fire Stand
    廣播訊息 win!
    等待 1 秒
    造型換成 Big Slide
    變數 Movements Frozen 設為 1
    重複直到 y 座標 < -75
        y 改變 -5
    等待 0.5 秒
    面朝 -90 度
    x 改變 20
    等待 0.5 秒
    面朝 90 度
    造型換成 Big Jump
    變數 Xv 設為 4
    變數 Yv 設為 12
    重複直到 y 座標 = -100
        變數 Yv 改變 -1.7
        y 改變 Yv
        x 改變 Xv
        如果 y 座標 < -100 那麼
            y 設為 -100
    造型換成 Big Stand
    否則
        面朝 90 度
        造型換成 Fire Slide
        變數 Movements Frozen 設為 1
        重複直到 y 座標 < -75
            y 改變 -5
        等待 0.5 秒
        面朝 -90 度
        x 改變 20
        等待 0.5 秒
        面朝 90 度
        造型換成 Fire Jump
        變數 Xv 設為 4
        變數 Yv 設為 12
        重複直到 y 座標 = -100
            變數 Yv 改變 -1.7
            y 改變 Yv
            x 改變 Xv
            如果 y 座標 < -100 那麼
                y 設為 -100
        造型換成 Fire Stand
    廣播訊息 win!
    等待 1 秒
    造型換成 Stand

```

通關動畫

➤➤➤ All Tiles-程式碼

遊戲重新開始 · reset

當前Level關卡資料
(擴充性展示)

產生多個分身磚塊

```

當收到訊息 restart level
分身刪除
  
```

```

定義 upload level
刪除 Actual Level Edited by Game 的 第 all 項
變數 scroll# 設為 1
重複 清單 Level OuO 的長度 次
  添加 Level OuO 的 第 scroll# 項 到 Actual Level Edited by Game
  變數 scroll# 改變 1
  
```

```

定義 Make Clonies Quick
筆跡全部清除
造型換成 b
定位到 x: -224 y: 160
重複 13 次
  重複 17 次
    建立 自己 的分身
    移動 27.5 點
  y 改變 -27.5
  x 設為 -224
  
```

```

當收到訊息 restart level
等待 0.1 秒
音量設為 60 %
變數 indiv.is a clone? 設為 0
隱藏
刪除 >> TYPE 的 第 all 項
刪除 >> X POS 的 第 all 項
刪除 >> Y POS 的 第 all 項
刪除 >> XV 的 第 all 項
刪除 >> YV 的 第 all 項
刪除 >> TOP HB 的 第 all 項
刪除 >> BOTTOM HB 的 第 all 項
刪除 >> RIGHT HB 的 第 all 項
刪除 >> LEFT HB 的 第 all 項
刪除 >> SHOULD TERMINATE? 的 第 all 項
刪除 >> DIR 的 第 all 項
變數 Global Next ID# 設為 1
Make Clonies Quick
等待 0 秒
廣播訊息 official start!
等待 0.3 秒
upload level
  
```

遊戲開始時, reset
初始化分身, 刪除暫存障
礙物資料, 重新讀取關卡
數據。
重設全域變數

》》》 All Tiles-程式碼

當建立一個新的分身時，把新分身的屬性加到各項清單裡
之後遊戲邏輯就可以透過清單索引，逐步讀取或修改每個分身的狀態。

```

定義 Create New
  添加 2 到 >> TYPE
  添加 x 座標 到 >> X POS
  添加 y 座標 到 >> Y POS
  添加 0 到 >> XV
  添加 0 到 >> YV
  添加 18 到 >> TOP HB
  添加 -8 到 >> BOTTOM HB
  添加 13 到 >> RIGHT HB
  添加 -13 到 >> LEFT HB
  添加 0 到 >> SHOULD TERMINATE?
  添加 90 到 >> DIR
  
```

```

當收到訊息 official start!
  如果 indv. is a clone? = 1 那麼
    重複無限次
      等待 0.125 秒
      變數 indv. [?] Rotation 設為 2
      等待 0.125 秒
      變數 indv. [?] Rotation 設為 3
      等待 0.125 秒
      變數 indv. [?] Rotation 設為 2
      等待 0.125 秒
      變數 indv. [?] Rotation 設為 1
      等待 0.125 秒
  
```

方塊分身初始的「閃爍 / 旋轉」預覽效果。

```

當分身產生
  如果 造型 編號 = 5 不成立 那麼
    變數 Indv List # 設為 Global Next ID#
    變數 Global Next ID# 改變 1
    變數 indv. is a clone? 設為 1
    變數 Indv. needs to be tryna delete 設為 0
    Create New
    造型換成 b
    顯示
    變數 X Default Position 設為 x 座標
    變數 Indv Column Number 設為 x 座標 + 224 / 27.5 + 1
    尺寸設為 86 %
    造型換成 g
    變數 indv. [?] already hit? 設為 0
    變數 indv. [?] Rotation 設為 1
    變數 indv. supposed to be Y pos 設為 y 座標
    變數 indv. extra y after bumped 設為 0
    變數 indv. how many coins left 設為 5
  
```

分配一個全域唯一 ID 作為在所有清單裡的索引。標記自己是「有效分身」設置視覺，存儲額外屬性，後續控制各種碰撞、旋轉、彈起與金幣踩踏次數的行為。

➤➤➤ All Tiles-程式碼

根據不同造型，
有不同的碰撞
事件反應
Ex 實心磚塊，
問號磚塊，金
幣磚塊等等

```

當收到訊息 official start
如果 indiv is a clone? = 1 那麼
  重複無限次
    y 設為 indiv, supposed to be Y pos + indiv, extra y after bumped
    造型換成 字符串 Actual Level Edited by Game 的第 四捨五入數值 y 座標 - 100 / -27 + 1 項 的第 SCROLL X * -1 + Indiv, Column Number 字
    顯示
    如果 造型編號 = 2 不成立 那麼
      如果 造型編號 = 7 或 造型編號 = 12 那麼
        替換 >> TOP HB 的第 Indiv List # 項為 42
        替換 >> BOTTOM HB 的第 Indiv List # 項為 -8
        替換 >> LEFT HB 的第 Indiv List # 項為 -15
        替換 >> RIGHT HB 的第 Indiv List # 項為 38
      否則
        如果 造型編號 = 9 或 造型編號 = 10 或 造型編號 = 11 或 造型編號 = 17 那麼
          替換 >> TOP HB 的第 Indiv List # 項為 -10
          替換 >> BOTTOM HB 的第 Indiv List # 項為 10
          替換 >> LEFT HB 的第 Indiv List # 項為 10
          替換 >> RIGHT HB 的第 Indiv List # 項為 -10
        否則
          如果 造型編號 = 8 那麼
            如果 Still able to spawn enemy? = 1 那麼
              造型換成 g
              替換 >> TOP HB 的第 Indiv List # 項為 -10
              替換 >> BOTTOM HB 的第 Indiv List # 項為 10
              替換 >> LEFT HB 的第 Indiv List # 項為 10
              替換 >> RIGHT HB 的第 Indiv List # 項為 -10
              替換 >> X POS 的第 Indiv List # 項為 x 座標
              替換 >> Y POS 的第 Indiv List # 項為 indiv, supposed to be Y pos
            替換 Replace block of row(t to b) Actual Level Edited by Game 的第 四捨五入數值 indiv, supposed to be Y pos - 100 / -27 + 1 項 column 無條件捨去 數值 SCROLL X * -1 + Indiv, Column Number with g
            變數 Spawn Y Pos 設為 indiv, supposed to be Y pos
            變數 Still able to spawn enemy? 設為 0
            廣播訊息 spawn goomba
          隱藏
  
```

➤➤➤ All Tiles-程式碼

磚塊的碰撞狀態判斷
function

```

定義 Sense for Mario
變數 Indiv_Touching What?? 設為 0
如果 >> X POC 的 Indiv List # 項 >> RIGHT HD 的 Indiv List # 項 > Mario 的 X 座標 + Horizontal HDBox 且 >> X POC 的 Indiv List # 項 >> LEFT HD 的 Indiv List # 項 < Mario 的 X 座標
    變數 Indiv_Touching What?? 設為 1

定義 Replace block of row(i to b) row# column column# with replacement
變數 scroll# 設為 1
重複執行 1 次
    變數 row# 設為 Actual Level Edited by Game 的 scroll# 項
    變數 scroll# 改變 1
    變數 indiv row# 設為 scroll#
    變數 Substring 設為 0
    變數 scroll# 設為 1
    重複執行 1 次
        變數 column# 設為 1
        變數 Substring 設為 字符串 Substring 字符串 Actual Level Edited by Game 的 indiv row# 項的 scroll# 字
        變數 scroll# 改變 1
        變數 Substring 設為 字符串 Substring replacement
        變數 scroll# 改變 1
        變數 字符串 Actual Level Edited by Game 的 indiv row# 項的長度 - column# 次
        變數 Substring 設為 字符串 Substring 字符串 Actual Level Edited by Game 的 indiv row# 項的 scroll# 字
        變數 scroll# 改變 1
    替換 Actual Level Edited by Game 的 indiv row# 項為 Substring
    
```

動態更新關卡目
前狀態

Pen rendererror-程式碼



確保顯示在最上層，遮住渲染中的地圖

Other Clones-程式碼

當收到訊息要產生
goomba(絕緣體)時對
應的動作

```

當收到訊息 spawn goomba
如果 SCROLL X < Previous Spawned Enemy Scroll X 那麼
  如果 indiv. is a clone? = 0 那麼
    定位到 x: 234 y: Spawn Y Pos
    地型換成 Goomba
    建立 自己 的分身
    變數 Previous Spawned Enemy Scroll X 設為 SCROLL X
  
```

```

定義 Create New
  添加 x 座標 到 >> X POS
  添加 y 座標 到 >> Y POS
  添加 0 到 >> XV
  添加 0 到 >> YV
  添加 90 到 >> DIR
  添加 0 到 >> SHOULD TERMINATE?

```

創造分身function

遊戲重新開始 · reset

```

當收到訊息 restart level
  分身刪除

當收到訊息 official start!
  分身刪除

當收到訊息 official start!
  等待 0.1 秒
  變數 indiv. is a clone? 設為 0
  隱藏
  音量設為 60 %
  廣播訊息 spawn flag
  面朝 90 度
  尺寸設為 100 %

```

Other Clones-程式碼

```

定義 SENSE (Goomba/Meshroom)
變數 scroll# 改為 1
變數 indiv is touching something 改為 0
變數 Condition of block underneath 改為 0
重複 清單 >> DIR 的長度 次
如果 >> TYPE 的 scroll# 項 = 2 或 >> TYPE 的 scroll# 項 = 92 或 >> TYPE 的 scroll# 項 = 100 或 設置 編號 = 4 且 >> TYPE 的 scroll# 項 = 91 不成立 那麼
如果 scroll# = Indiv List# 不成立 那麼
如果 >> X POS 的 scroll# 項 > >> RIGHT HB 的 scroll# 項 > > X POS 的 Indiv List# 項 + >> LEFT HB 的 Indiv List# 項 且 >> X POS 的 scroll# 項 > >> LEFT HB 的 scroll# 項 < >> X POS 的 Indiv List# 項 + >> RIGHT HB 的 Indiv List# 項 且 >> Y POS 的 scroll# 項 > >> TOP HB 的 scroll# 項
如果 >> TYPE 的 scroll# 項 = 93 或 >> TYPE 的 scroll# 項 = 95 那麼
變數 indiv is touching something 改為 2
替換 >> DIR 的 scroll# 項為 111
否則
如果 indiv is touching something = 2 不成立 那麼
如果 >> TYPE 的 Indiv List# 項 = 95 且 >> TYPE 的 scroll# 項 = 91 或 >> TYPE 的 scroll# 項 = 94 那麼
否則
變數 indiv is touching something 改為 1
如果 >> DIR 的 scroll# 項 = 12345 那麼
變數 Condition of block underneath 改為 1
變數 scroll# 改為 1

```

掃描整張地圖的每一格 tile，並判斷「這隻敵人 / 道具分身 (indiv) 目前：是否正好跟 Mario 碰撞腳底下是否有方塊 (用來實作掉落、重力或碰撞反彈) 若撞到特殊方塊 (如問號箱)，要標記方塊並改變它的狀態

Other Clones-程式碼

在問號箱或磚塊被撞碎時，生成四塊碎片，模擬方塊破裂

```

當收到訊息 spawn block pieces
如果 indv. is a clone? = 0 那麼
  定位到 x: Spawn X Pos y: Spawn Y Pos
  x 改變 6
  y 改變 6
  造型換成 block piece
  變數 Indiv. Xv 設為 3
  建立 自己 的分身
  y 改變 -13
  建立 自己 的分身
  x 改變 -13
  變數 Indiv. Xv 設為 -3
  建立 自己 的分身
  y 改變 13
  建立 自己 的分身
  
```

在關卡終點或特殊位置生成終點旗子分身

```

當分身產生
變數 indv. is a clone? 設為 1

當收到訊息 spawn flag
如果 indv. is a clone? = 0 那麼
  定位到 x: Spawn X Pos y: Spawn Y Pos
  造型換成 flag
  建立 自己 的分身
  
```

問號箱裡彈出蘑菇(磷原子)或火焰花(As)的邏輯

```

當收到訊息 spawn mushroom
如果 indv. is a clone? = 0 那麼
  定位到 x: Spawn X Pos y: Spawn Y Pos
  如果 MARIO'S TYPE = 1 那麼
    造型換成 Mushroom
  否則
    造型換成 Fireflower
  建立 自己 的分身
  
```

取得火焰花(As摻雜)的 Mario 按下射火球鍵時，從頭頂生成火球(電子)

```

當收到訊息 shoot fireball
如果 indv. is a clone? = 0 那麼
  定位到 Mario 位置
  造型換成 Fireball
  建立 自己 的分身
  
```

Other Clones-程式碼

設定碎片方塊一開始的大小、旋轉、速度。模擬向下的物理效果，達到邊界後刪除

```

    當分身產生
    如果 造型 編號 = 3 那麼
        顯示
        圖層移到 最上 層
        變數 Indiv_Yv 設為 9
        變數 Indiv_is_a_clone? 設為 1
        尺寸設為 60 %
        y 改變 5
        重複無限次
            右轉 45 度
            重複 5 次
                變數 Indiv_Yv 改變 -1
                變數 Indiv_Xv 設為 Indiv_Xv * 0.98
                如果 Mario 的 x 座標 < 0 那麼
                    x 改變 Indiv_Xv
                否則
                    x 改變 Indiv_Xv - Xv
                y 改變 Indiv_Yv
                圖層移到 最上 層
                如果 Indiv_Yv < -12 那麼
                    變數 Indiv_Yv 設為 -12
                如果 碰到 邊緣? 那麼
                    分身刪除
            結束重複
        結束重複
    如果 造型 編號 = 4 那麼
        播放音效 smb_powerup_appears
        顯示
        圖層移到 最上 層
        變數 Indiv_List # 設為 Global_Next_ID#
        變數 Global_Next_ID# 改變 1
        變數 Indiv_is_a_clone? 設為 1
        變數 Indiv_needs_to_be_tryna_delete 設為 0
        Create New
        添加 92 到 >> TYPE
        添加 14 到 >> TOP_HB
        添加 -10 到 >> BOTTOM_HB
        添加 12 到 >> RIGHT_HB
        添加 -12 到 >> LEFT_HB
        變數 Indiv_Xv 設為 3
        變數 Indiv_Yv 設為 0
        y 改變 15
        重複 25 次
            y 改變 1.2
            圖層 下 移 100 層
            如果 Mario 的 x 座標 < 0 那麼
                播放音效 smb_powerup_appears
            否則
                x 改變 Xv * -0.85
        結束重複
        替換 >> Y_POS 的第 Indiv_List # 項為 y 座標
        替換 >> X_POS 的第 Indiv_List # 項為 x 座標
        重複無限次
            變數 Indiv_Yv 改變 -1.5
    結束重複
    
```

蘑菇(磷原子)的出生動畫先往上跑之後往右移動，碰到障礙物反彈

Intro-程式碼

確保Intro畫面不被阻擋

```
當收到訊息 start
造型換成 Goodbye, thanks! :D
顯示
圖像效果清除
圖像效果 ghost 設為 100
定位到 x: 0 y: 0
變數 Intro Already Been Played 設為 1

當收到訊息 make sure vars are in front
圖層移到 最上 層
```

The code blocks are arranged in two columns. The left column contains: '當收到訊息 start', '造型換成 Goodbye, thanks! :D', '顯示', '圖像效果清除', '圖像效果 ghost 設為 100', '定位到 x: 0 y: 0', and '變數 Intro Already Been Played 設為 1'. The right column contains: '當 被點擊', '計時器重置', '圖層移到 最上 層', and '廣播訊息 start'. Below these, there is a separate block: '當收到訊息 make sure vars are in front' followed by '圖層移到 最上 層'.



Scratch

半導體
瑪利歐

Rabboni

感謝聆聽!