

Swift Playgrounds

王昱景 Brian Wang

ycwang812.at.tw

brian.wang.frontline@gmail.com

What is ...

- Swift Playgrounds
<https://www.apple.com/tw/swift/playgrounds/>
- Swift
<https://www.apple.com/tw/swift/>
- Xcode

Swift Playgrounds 課程

[新增 Playground](#)

哈囉，Byte

Apple

新手

Swift 4.2 Edition

取得

開始你的程式設計之旅

描述

跟神奇的新朋友 Byte 打聲招呼吧！這項挑戰設計為一小時完成，它包含「學習程式設計 1」課程中的部分關卡。你將在一個動態的 3D 世界中突破重重關卡，學習如何發出簡單指令、編寫函數以及執行迴圈。在這些關卡中，你可以透過編寫真正的程式碼，探索 Swift 的基礎知識。程式設計者也是使用相同的程式碼。

新功能

- 錯誤修正

資訊

發行者： Apple

版本： 4.0.1

發行日期： 2018年8月30日

版本： Swift 4.2 版

語言： 英文、巴西葡萄牙文、簡體中文、繁體中文、荷蘭文、法文、德文、義大利文、日文、韓文、拉丁美洲西班牙文、西班牙文、瑞典文、泰文和土耳其文。

< 新增 Playground



學習程式設計 1

Apple

Swift 4.2 Edition

取得

Swift 基礎

You'll start by writing **commands** to move a character named Byte around a puzzle world, performing tasks.

For example, you'll tell Byte to move forward:

```
moveForward()
```

< Issuing Commands >

Goal: Use Swift commands to tell Byte to move and collect a gem.

Your character, Byte, loves to collect gems but can't do it alone. In this first puzzle, you'll need to write Swift **commands** to move Byte across the puzzle world to collect a gem.

1. Look for the gem in the puzzle world.
2. Enter the correct combination of the `moveForward()` and `collectGem()` commands.
3. Tap Run My Code.

Tap to enter code

< Using Loops >

Challenge: Use an `if` statement to trigger a sequence of commands if your character is on a gem.

Congratulations! You've learned how to write **conditional code** using `if` statements and `else if` blocks.

A condition like `isOnGem` is always either true or false. This is known as a Boolean value. Coders often use Boolean values with **conditional code** to tell a program when to run certain blocks of code.

1. In the `if` statement below, use the Boolean condition `isOnGem` and add commands to run if the condition is true.
2. Modify or keep the existing `else` block to run code if your Boolean condition is false.
3. If necessary, tweak the number of times your `for` loop runs.

```
for _ in 1 ... 10 {
    if isOnGem {
        moveForward()
    }
    else {
        moveForward()
    }
}
```

< Conditional Climb >

Challenge: Use an `if` statement to trigger a sequence of commands if your character is on a gem.

Congratulations! You've learned how to write **conditional code** using `if` statements and `else if` blocks.

A condition like `isOnGem` is always either true or false. This is known as a Boolean value. Coders often use Boolean values with **conditional code** to tell a program when to run certain blocks of code.

1. In the `if` statement below, use the Boolean condition `isOnGem` and add commands to run if the condition is true.
2. Modify or keep the existing `else` block to run code if your Boolean condition is false.
3. If necessary, tweak the number of times your `for` loop runs.

```
for _ in 1 ... 10 {
    if isOnGem {
        moveForward()
    }
    else {
        moveForward()
    }
}
```

描述

準備好掌握程式碼的基本知識了嗎？在「學習程式設計 1」中，你將探索 Swift 的基本知識，這是一款為 Apple 產品用來製作 App 的程式設計語言。

透過在動態 3D 關卡世界中通關，你將發展出一組程式設計的技能，建構你的基礎程式設計詞彙。程式設計之旅將從簡單的指令、函數和迴圈開始。你一開始就會編寫真實的程式碼，也是真正的程式設計者所使用的程式碼。

跟隨學習之旅的腳步，你會接觸到能回應環境變化的條件碼，然後學習運算子和 while 迴圈，讓程式碼變得更精確、更易於重複使用。融會貫通各種技巧後，你就能寫出簡單而強大的演算法，用相同的程式碼通過複雜的關卡。

< 新增 Playground



學習程式設計 2

Apple

Swift 4.2 Edition

[取得](#)

進階內容

Here's how you place a new block.

1. Create an instance of type **Block**.

```
let block = Block()
```

2. Use the **place()** method, and pass in your arguments.

```
world.place(block, atColumn: 5, row: 0)
```



Goal: Collect exactly seven gems.

You've learned to use a **variable** to keep track of a changing value, incrementing it when needed. In this new puzzle, you'll use this knowledge to collect exactly seven gems. The gems appear not only in random locations, but also at random times.

To solve the puzzle, you'll need to use a **while** loop with a **Boolean** condition that returns **false** after you've collected all seven gems. You'll use a **comparison operator** (**==**) to compare the **gemCounter** value to the **int** value of 7.

Using a comparison operator
while gemCounter < 7 {
}
1 Declare your **gemCounter** variable and set the value to 0.
2 Increment the **gemCounter** value each time your character collects a gem.
3 Use a **while** loop to keep collecting gems until all seven are collected.



描述

在「學習程式設計 2」中，你將積累 Swift 的基礎知識。旅程從簡單的通關開始，進階到建構你自己的世界。

你將學習變數和類型，這是可儲存和取用資訊的程式設計結構。編寫程式碼時，你可以結合這些新技巧、初始化和參數，透過更多方式與角色和關卡世界互動，改變世界本身的規則。此外，你還會學習陣列，透過它建立任何類型的列表，形成易於管理、儲存和放置的複雜項目序列，讓你創建的關卡世界更加豐富。

新功能

- 修復問題

< 新增 Playground



學習程式設計 3

探索宇宙

Apple

Swift 4.2 Edition

取得

探索宇宙

In Learn to Code 3, you'll use code to fill my universe with animated images and text.

3 of 7

Create: Interstellar Symmetry

You can place several images on the scene each time you move your finger and `addImage()` is called. If you place them in mirror-image positions around the scene, you can trace cool, kaleidoscope-like patterns.

Each time you place an image at your finger position in one quadrant of the scene, place three more in mirror-image positions in the other three quadrants. To work out the positions of the images, start with the `absolute` value of the x and y touch coordinates. The symmetrical positions in the four quadrants are then top-right (`x, y`), bottom-right (`-x, -y`), bottom-left (`-x, y`), and top-left (`x, -y`). The code below creates four `Graphic` instances, each with the same random animal image, and puts them into the `graphics` array.

Place one graphic from the `graphics` array in each quadrant of the scene. The `abs()` function gets the absolute x and y values; use them to get the x and y coordinates for each quadrant.

Ideas:

- Set scale and rotation for each graphic to a random `Double` value.
- Have each graphic gradually disappear after it's been placed, by calling one of these graphic methods: `fadeOut()`, `spinAndPop()`, or `wirlIMay()`.
- Instead of just placing each graphic at its final position, first place it at `touch.position` and then use `moveToDuration()` to animate it into place.

When you're finished, move on to the [next page](#).

```

scene.backgroundColor = black
let animals = ["cow", "pig", "sheep", "goat", "bear"]
let lastPlacePosition = Point(x: 0, y: 0)

func addImage(touch: Touch) {
    // Space out the graphics.
    let angle = Double.pi * 2 * random(0, 1)
    let x = touch.x + cos(angle) * 100
    let y = touch.y + sin(angle) * 100
    let absX = abs(x)
    let absY = abs(y)
    let quadrant = (x > 0) ? (y > 0) ? "top-right" : "bottom-right" : (y > 0) ? "top-left" : "bottom-left"
    let angle = (quadrant == "top-right") ? 0 : (quadrant == "bottom-right") ? 180 : 90 : 270
    let scale = random(0.5, 2.0)
    let rotation = random(0, 360)
    let graphic = Graphic(image: animals.random())
    graphic.setPosition(x: x, y: y)
    graphic.setScale(scale)
    graphic.setRotation(rotation)
    graphic.setAngle(angle)
    graphic.fadeOut()
    graphic.spinAndPop()
    graphic.wirlIMay()
    let graphics = [graphic]
    graphics.append(Graphic(image: animals.random()))
    graphics.append(Graphic(image: animals.random())))
    scene.place(graphics, at: Point(x: 0, y: 0))
}

```

Placing Text

The `scene.write()` function only puts lines of plain text on the scene. Not much fun if you want to do something more adventurous, like putting text in a particular place, or making it look stylized.

Fortunately, you can create a `Graphic` with text and place it on the scene yourself, just as you would an image graphic.

Creating a text graphic:

```

let graphic = Graphic(text: "Place me!")

```

Placing a text graphic:

```

scene.place(graphic, at: Point(x: 300, y: 300))

```

Goal: Place strings on the scene.

The `scene.write()` function only puts lines of plain text on the scene. Not much fun if you want to do something more adventurous, like putting text in a particular place, or making it look stylized.

Fortunately, you can create a `Graphic` with text and place it on the scene yourself, just as you would an image graphic.

Ideas:

- For each caption, create a `Graphic` instance with the caption.
- Place each graphic on the scene using a `Point` instance for the x, y coordinates.

```

scene.clear()
scene.backgroundImage = nil

// Use your own captions.
let caption1 = "Ooowheeee"
let caption2 = "Blurpl! Gloomk."
let caption3 = "Grrrrr!"
let caption4 = "I'm friendly!"

// Add captions to scene.
let graphic1 = Graphic(text: caption1)
scene.place(graphic1, at: Point(x: -250, y: 320))
let graphic2 = Graphic(text: caption2)
scene.place(graphic2, at: Point(x: 250, y: 320))

```

Well done!

Idea: Set the background image to your own photo—maybe a selfie with your friends—and put some captions on it!

[Next Page](#)

Removing a Graphic

Goal: Remove a graphic by touching it.

Blu's universe has a problem with UFFs—Unidentified Flying Fructoids. They buzz around in weird orbits and get in the way of everything.

Your task is to create a `Squish` tool to destroy the UFFs by touching them. But they're tough-skinned, and you have to hit them several times before they get pulled into the black hole.

- Add code to `squishGraphic()` to reduce the `scale` value of the touched graphic by 0.5.
- Add code to reduce the `alpha` value by 0.25. Run the code, add some UFFs, and try tapping them. They should get smaller and fade away.
- Now add code to check if the scale of the touched graphic is less than 0.6. If it is, call the `moveAndZap()` method of the graphic to move it to a new position (the black hole), and then remove it from the scene.
- Run the code, add a bunch of UFFs, and start squishing them!

```

// Squish tool event handler.
func squishGraphic(graphic: Graphic) {
    // Add code here.
}

// UFF tool event handler.

```

描述

在「學習程式設計 3」中，你將擴展你的程式設計技能，思考方式會更像一位 App 開發者。

你將接觸 Blu 宇宙的星際空間，探索專業開發者們所運用的強大程式設計理念，建構一組創意工具。當你學習圖形和座標時，會進而瞭解如何放置和處理圖形。結合這些技巧與觸碰事件，在星際空間中繪製各種藝術形狀。

在快速地瞭解字串之後，掌握在太空中放入字詞甚至是表情符號。接著，加入語音合成器和聲音效果，讓你的聲音在寂靜的 Blu 宇宙中響起。

最後，使用真實事件（例如移動手指或點按）啟動你的程式碼，探索事件處理器。你將使用事件處理器為外星人編舞，讓宇宙變成巨型的樂器，或建構自己的數位鎖。完成這些內容後，你就能熟練地結合掌握的各項技巧，編寫最棒的進階程式碼！

Jimu robot

- <https://ubtrobot.com/collections/jimu-robots>
- App
[https://itunes.apple.com/tw/app/jimu/
id1070681102?mt=8](https://itunes.apple.com/tw/app/jimu/id1070681102?mt=8)

< 搜尋

**Jimu**

UBTECH ROBOTICS CORP

打開

...

3.5 ★★★★☆

4份評分

4+

年齡

新功能

版本記錄

【硬件支持】

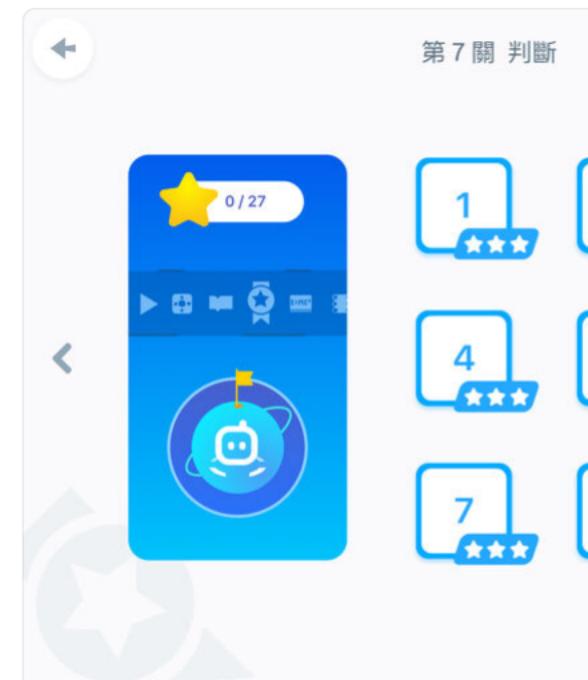
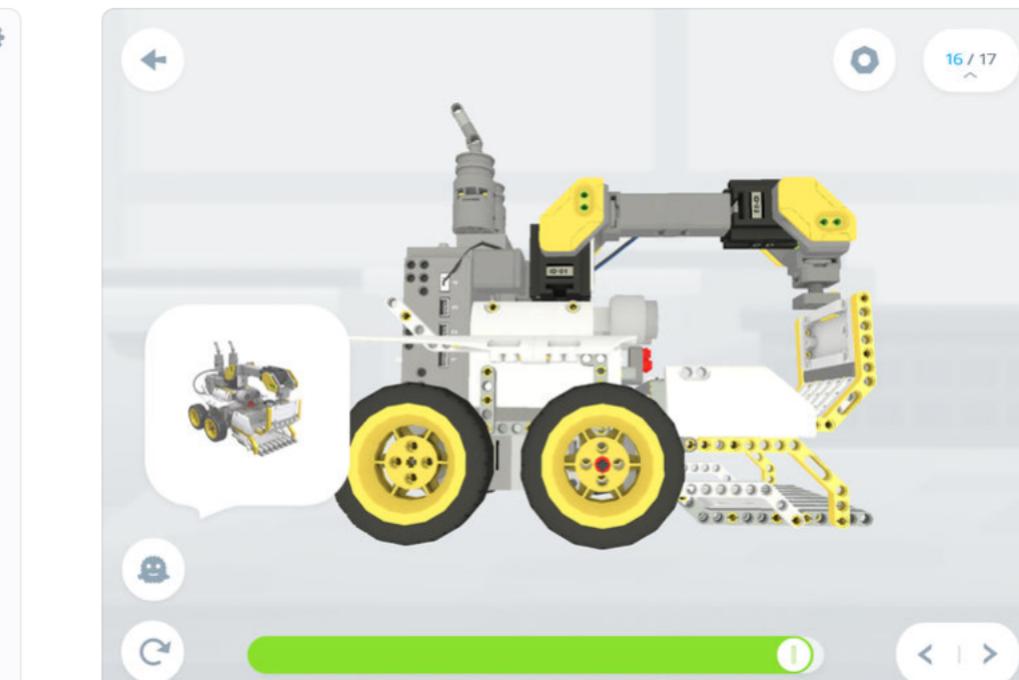
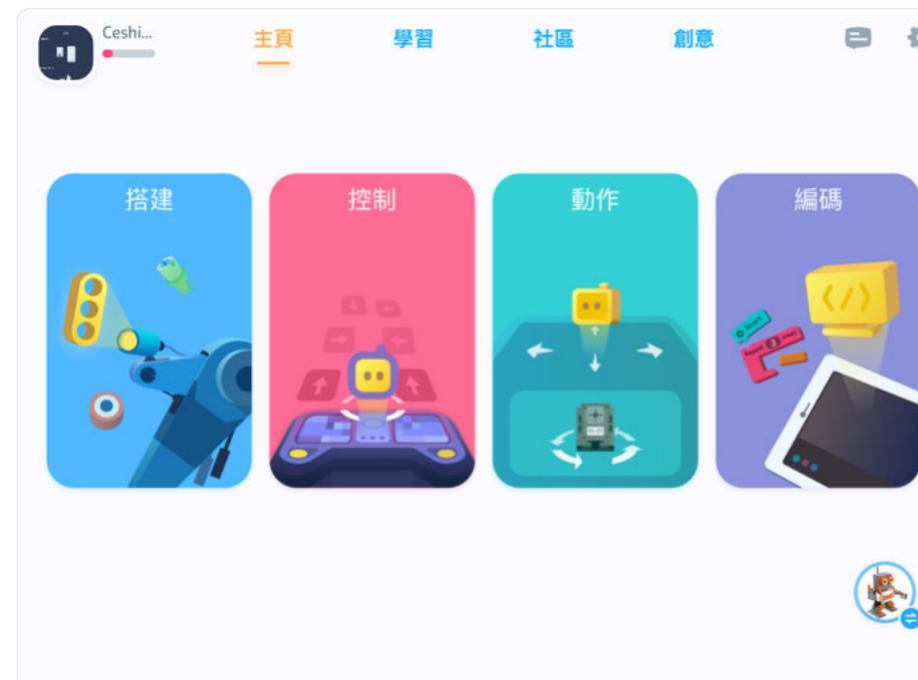
新增 Jimu 機器人“賽場先鋒”；

更多

2 天前

版本 3.2.0

預覽



連接件



C1x2



C2x2



C4x7



P09x2



P09x2



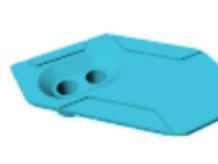
C5x6



C6x10



C8x2



P12x2



P12x2



MCx1



SERVOx6



C10x2



C11x3



P13x2

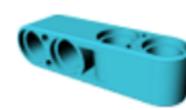
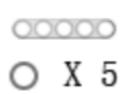


P13x2

关闭

裝飾件

2/3



P20x2

P22x1

P25x1

P26x3

P26x1

P34x2

P35x2



P35x6

P35x4

P35x6

P47x3

P48x48

P49x38

P54x2



P67x2

P68x2

P78x2

P80x1

关闭

連接線



关闭

Sphero BOLT

- <https://www.sphero.com>
- App
 - Sphero Play
[https://itunes.apple.com/tw/app/sphero-play/
id1280682522?mt=8](https://itunes.apple.com/tw/app/sphero-play/id1280682522?mt=8)
 - Sphero Edu
[https://itunes.apple.com/hk/app/sphero-edu/
id1017847674?mt=8](https://itunes.apple.com/hk/app/sphero-edu/id1017847674?mt=8)

< 搜尋



Sphero Play

Sphero, Inc.

打開

...

4.7 ★★★★★

49份評分

4+

年齡

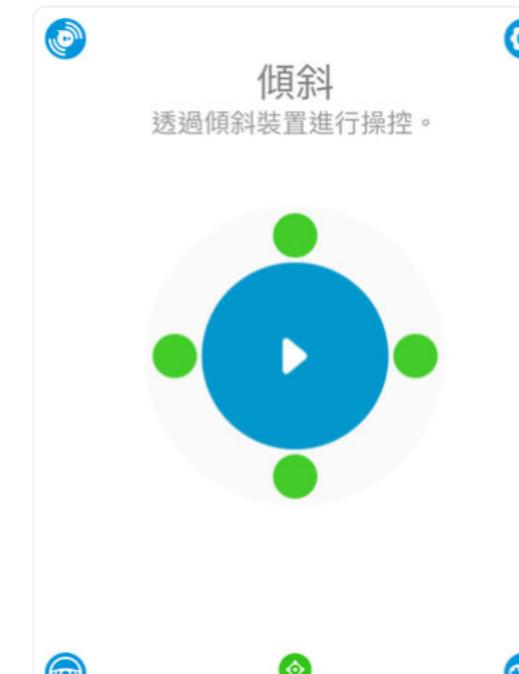
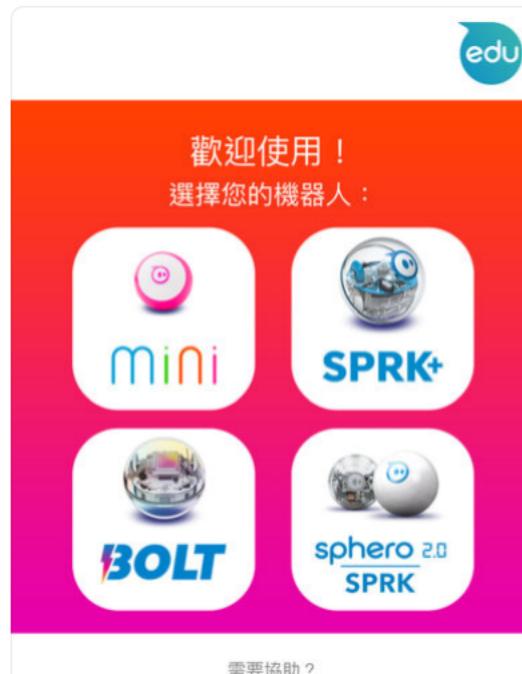
新功能

- Sphero 2.0 支援
- Sphero SPRK 支援

版本記錄

3 個月前
版本 2.3.0[更多](#)

預覽



< 搜尋



Sphero Edu

Sphero, Inc. Sphero Robots

打開

...

4.6 ★★★★★

12份評分

4+

年齡

新功能

版本記錄

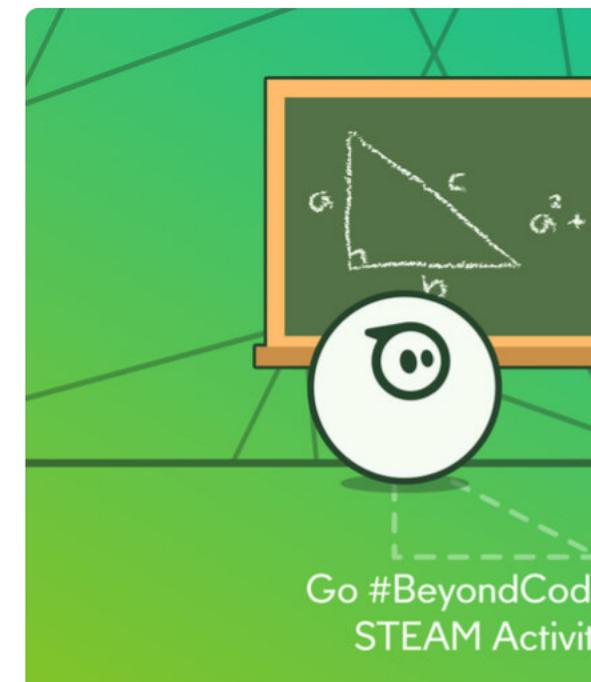
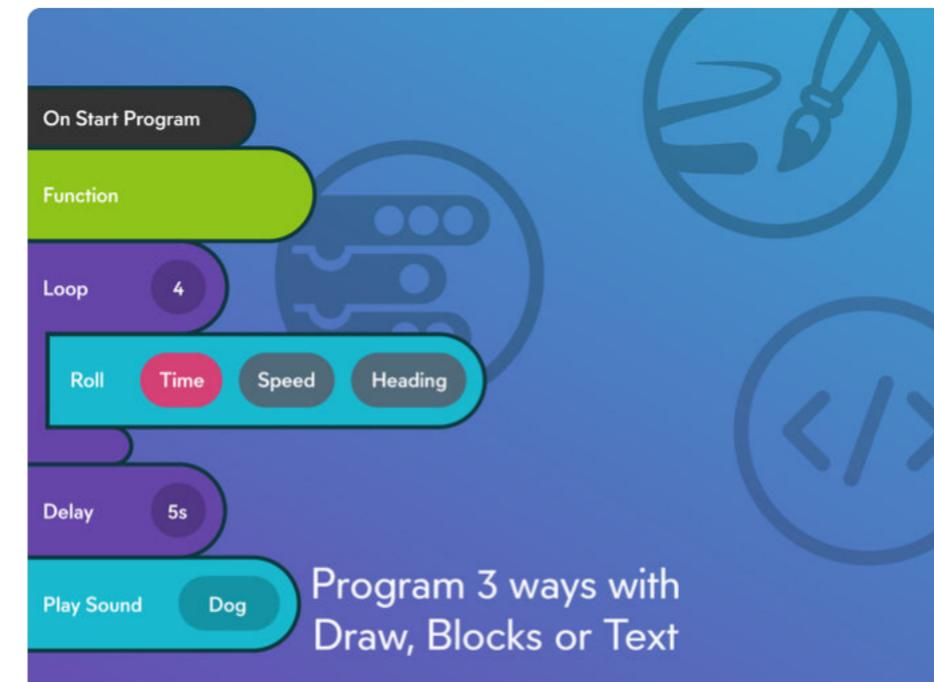
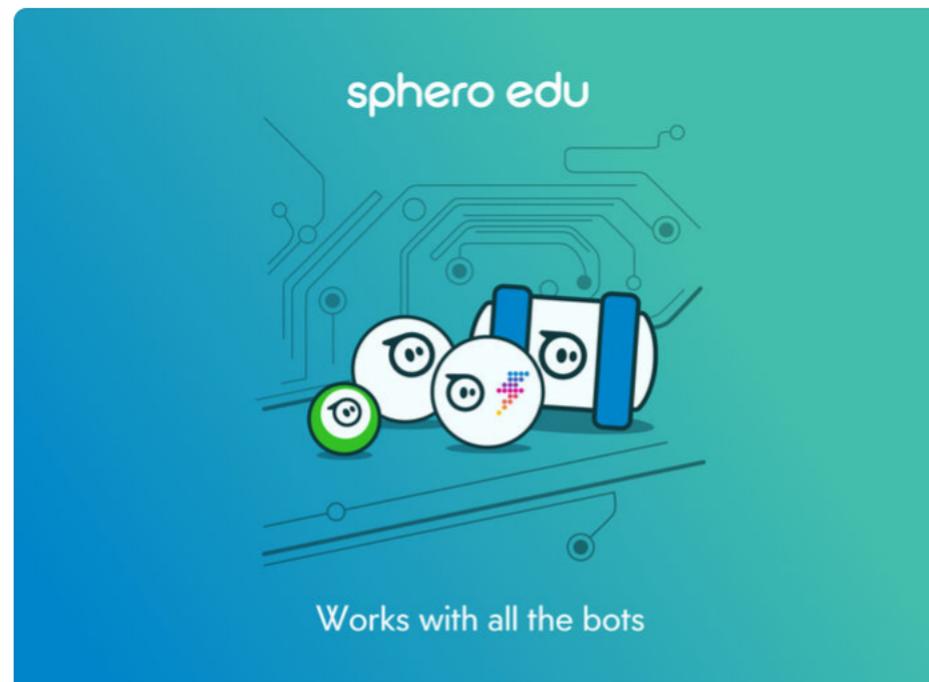
Holy smokes, Sphero BOLT is already getting upgrades!

- Infrared: Added IR broadcasting plus automatic following and evading behaviors so BOLT can get [更多](#)

6 個月前

版本 5.2.0

預覽



Parrot MAMBO FLY

- <https://www.parrot.com/global/>
- FreeFlight Mini
[https://itunes.apple.com/tw/app/freeflight-mini/
id1137022728?mt=8](https://itunes.apple.com/tw/app/freeflight-mini/id1137022728?mt=8)

< 搜尋



FreeFlight Mini

Parrot SA

打開

2.9 ★★★★☆

9份評分

4+

年齡



新功能

Full access to Facebook live feature.

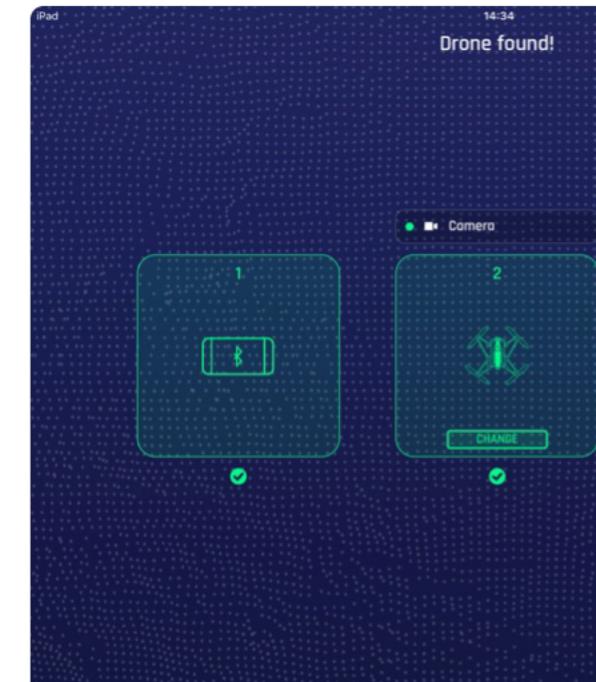
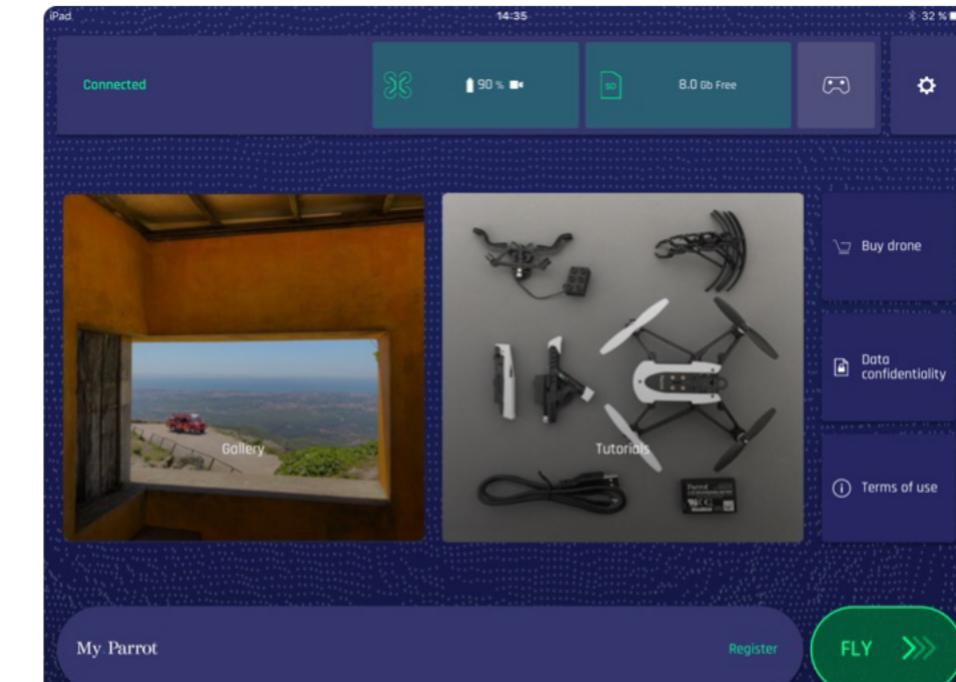
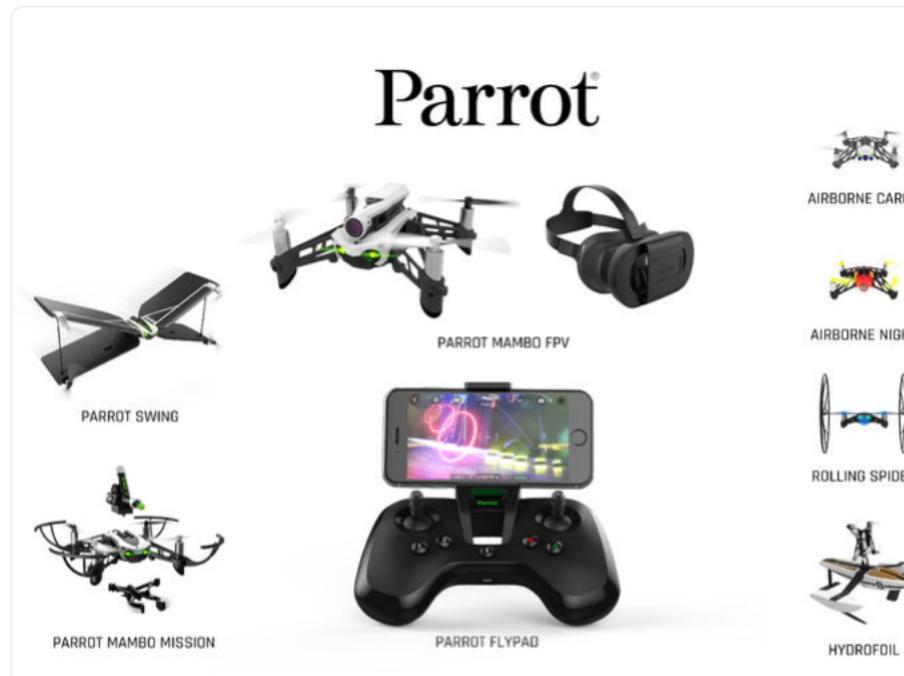
Correction of a bug in the drone firmware that blocked the wifi upgrade process.

[更多](#)

版本記錄

5 個月前
版本 5.5.8

預覽



在上課之前

- 檢查設備
- 韶體更新
- 充飽電源

- 講義、範例程式下載：
- <http://bit.ly/2KwwUsw>

