

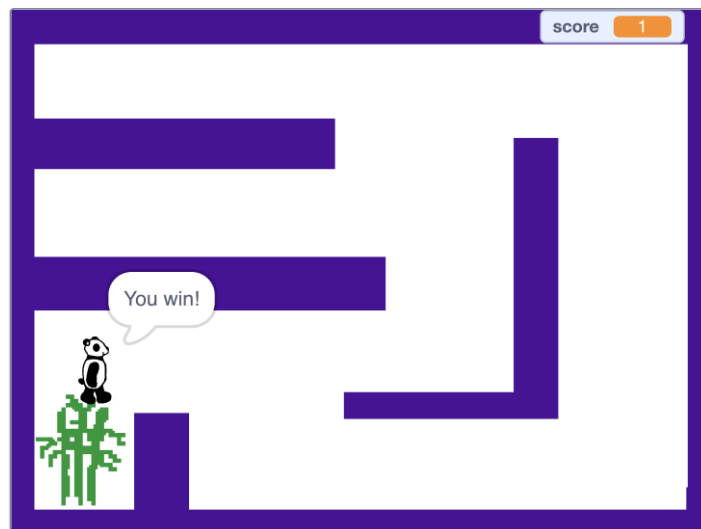
SCRATCH

迷宮遊戲

一起完成我們的遊戲吧！

在這個遊戲，你將利用鍵盤帶領熊貓避開迷宮的牆壁，前往竹葉林。到達終點將會得到 1 分，碰到牆壁將會扣 1 分。

在第二部分，你將會加入可以偵測熊貓有否碰到牆壁，或已經到達終點的指令方塊，以完成這個迷宮遊戲。你也將會為遊戲加上對白，分數和音效。

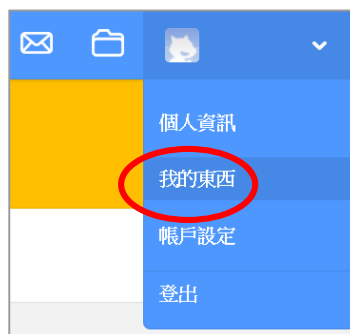


在這裏開始

❑ 在 scratch.mit.edu 登入你的帳戶。



❑ 點選 **My Stuff**，開啓你的迷宮遊戲專案。



❑ 點擊在網頁右上角的 **See inside** 按鈕，繼續製作你的遊戲。



❑ 在遊戲中，我們期望熊貓一觸碰到迷宮的牆壁，就會回到起點。所以，我們要在 **Control** 欄內拖放出 **if-then** 指令方塊。



SCRATCH

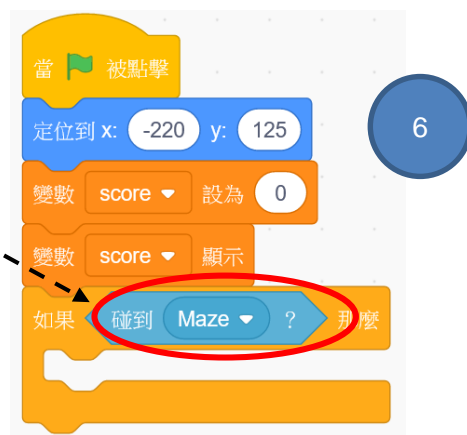
迷宮遊戲

我們繼續吧！

- 我們希望如果熊貓觸碰到迷宮牆壁，就會回到起點。所以要拉出 **Sensing** 欄內的 **touching Maze** 指令方塊。



- 把 **touching Maze** 指令方塊拖放到 **if-then** 指令方塊的空格中，然後把這方塊組合拖放到 **when green flag clicked** 指令方塊之下。現在，觸碰的變數 (即 **touching Maze**) 已設定為你所製作的迷宮的名稱 (**Maze**)。



- 把 **go to x: # y: #** 指令方塊複製並貼進 **if-then** 指令方塊內。



SCRATCH

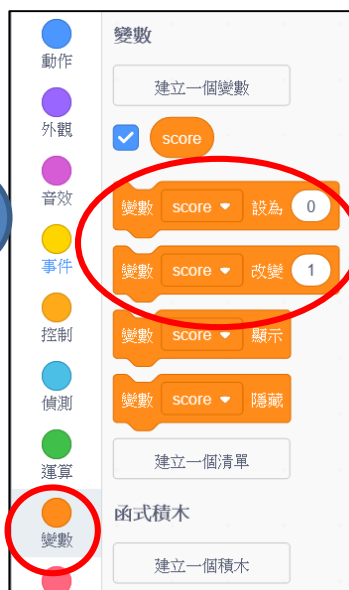
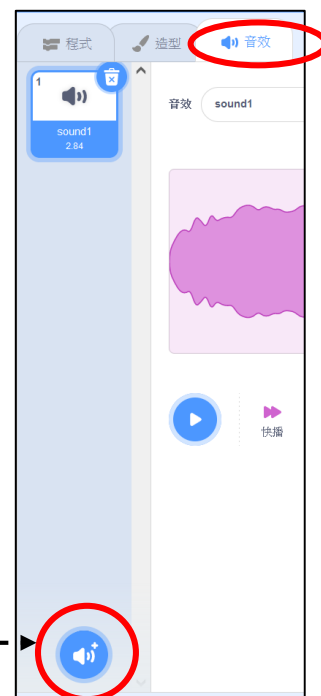
迷宮遊戲

我們繼續吧！

□ 適當加入額外的 **if-then** 指令方塊，令以下情況發生：

➤ 如果熊貓走到終點，牠會說：“You win!”，並播放聲音。

- 要令角色說話，先前往 **Looks** 欄。你會用哪個指令方塊？為甚麼？
- 要為專案加入聲音，先前往 **Sounds** 標籤，再點擊位於畫面左下方的 **Choose a Sound** 圖示，選擇聲音。
- 要讓角色發出播放想要的聲音，前往 **Code** 標籤中的 **Sound** 欄。



□ 熊貓找到竹林會加 1 分，觸碰到牆壁就會減 1 分。

提示：看看 **Variables** 欄內有甚麼可以用來改變分數。

□ 把熊貓角色彈回開始時的位置。

運算思維提示
if-then (如果—那麼) 是一個條件 (Conditional)。
「如果」滿足到這條件，
「那麼」事情就會發生。

SCRATCH

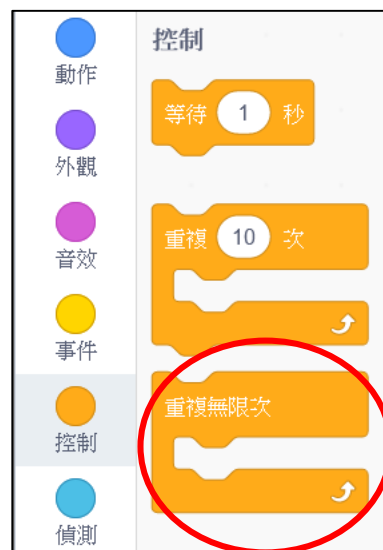
迷宮遊戲

快要完成了！

13 試玩一下你的遊戲。遊戲有正常運作嗎？如果沒有，你覺得遺漏了甚麼指令方塊？

14 重複無限次 (forever) 能令程式不斷檢查邊界事件 (boundary conditions)。在 **Control** 欄內找出 **forever** 指令方塊，然後試試把它加進你的程式。現在正常運作嗎？

15 如果你的遊戲還未能正常運作，可以請你的老師和同學幫助你除錯 (debug)。



聽從老師指示，把你的專案加到老師的 Studio 中：



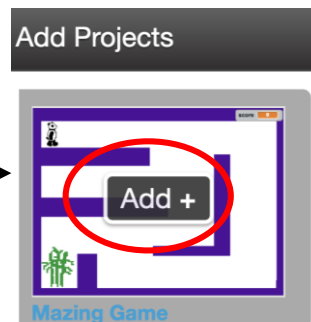
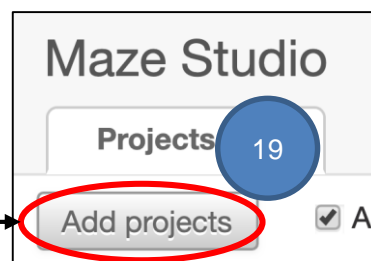
16 點擊 File menu 內的 **Save now** 按鈕，儲存你的專案。

17 點擊橙色的 **Share** 按鈕。

18 前往老師的 Maze Studio 中 (根據你收到的網址)。

點擊 **Add projects** 按鈕。

20 在頁面的下方選擇你的專案，加到 Studio 中。



SCRATCH

迷宮遊戲

運算思維概念

以下是我們在第二教節學會的運算思維概念。

L1U8.3 以SCRATCH開發迷宮遊戲

1. 序列：



2. 重複：



3. 條件：



SCRATCH

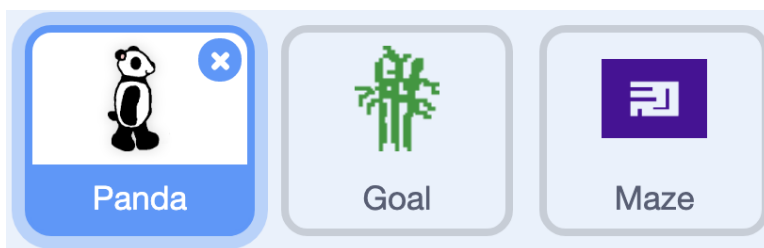
迷宮遊戲

運算思維概念

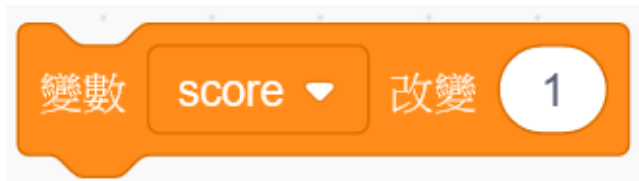
以下是我們在第二教節學會的運算思維概念。

L1U8.3 以SCRATCH開發迷宮遊戲

4. 命名：



5. 運算子：



SCRATCH

迷宮遊戲

運算思維實踐

以下是我們在本單元學會的運算思維實踐。

L1U8.3 以SCRATCH開發迷宮遊戲

1. 重用及整合：
 - a) 使用範例專案並整合它。
2. 反覆構思及漸進編程：
 - a) 增加指令方塊，控制角色移動；
 - b) 新增變數 score ；
 - c) 加入條件式 (if-then) 到專案；
 - d) 加入 forever (重複) 指令方塊。
3. 測試及除錯：
 - a) 測試碰撞牆壁並除錯；
 - b) 測試得分會否因應角色贏或輸而改變。
4. 算法思維：
 - a) 學會令角色根據按鈕的「事件」而移動。