

2 章

語音辨識篇

製作分辨聲音的數位寵物

你可以單憑聲音就知道對方是誰嗎？你應該能輕易分辨親朋好友的聲音吧！這次我們使用機器學習的語音辨識及分類結構，製作出可以分辨聲音並給予回應的數位寵物。完成當家人或朋友說「你好」時，可以回應對方「○○你好」的功能。這一章要使用「Teachable Machine」工具以及「TM2Scratch」擴充功能。

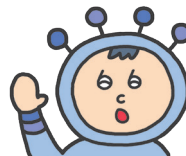


你跟它說說話！



「Shu 你好。」

你好～。



哇！它竟然認識我！



換我。小雞你好！



「Kikka 你好。」

真開心，它竟然認得我的聲音。



你們可以選擇自己喜歡的寵物，
讓它學習家人、朋友等各種人物的聲音，
一定很有趣喔！

這也是 Scratch 的擴充功能嗎？



是的！
這次使用了和第 1 章不同的擴充功能。

我已經迫不及待想試試看了！



2 編寫讓寵物給予回應的程式

接著要寫出讓小雞根據 Teachable Machine 的結果給予回應的程式。請按照以下方式組合積木。



使用的積木

- TM2Scratch → 接收到聲音標籤：「任何」
- 控制→如果「...」那麼
- TM2Scratch → 聲音「任何」被偵測？
- 文字轉語音→唸出「hello」

以「接收到聲音標籤：『任何』時」的事件積木開始，接著使用「如果『...』那麼」積木，按照各個類別改變回答。

利用「文字轉語音」擴充功能，加入「唸出『hello』」積木。



請試著說話，看看結果如何？會根據說話者給予回應嗎？

5 章

生成文章篇

讓 AI 創造遊戲角色的台詞

這一章我們要挑戰用熱門的生成式 AI（Generative AI）設計程式。在 Scratch 呼叫 ChatGPT 的擴充功能「ChatGPT2Scratch」，讓 AI 創造遊戲角色的台詞。我們該如何告訴 AI 呢？透過這一章的學習內容，你將能理解在程式設計加入生成式 AI 可以做到什麼。

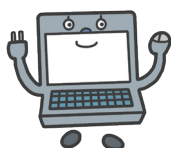


一般如何設計讓角色說話的程式呢？

哇！是飛龍與騎士！



我會使用 `說出 Hello! 持續 2 秒` 積木，設定攻擊時說出這句台詞，被打倒時說出那句台詞，在程式裡寫上希望角色說出來的內容。



沒錯。不過其實不用特地寫好台詞，先把設定與規則告訴 ChatGPT，它就能根據場景生成台詞。



可惡，竟然讓我吃到這種苦頭！
你這傢伙還挺厲害的嘛？



沒想到要用到這一招…
你看好了！這就是我真正的實力！

咦？飛龍會自己說話！？
不對，是 ChatGPT 幫我們創作了台詞吧！



沒錯。你可以像對話一樣，用一般的句子向 ChatGPT 下指令或提問。不過這需要一點技巧。

我想試試看！



聲明

ChatGPT 是利用 OpenAI 提供的 GPT 大型語言模型進行對話或溝通的應用程式名稱。本章介紹的 ChatGPT2Scratch 是可以使用 GPT 的擴充功能，因此「在程式中使用 GPT」或「在程式中使用像 ChatGPT 這種回覆機制」會是比較精準的描述，但是本書為了方便理解，刻意使用了「在程式中使用 ChatGPT」、「讓 ChatGPT 創作台詞」的描述。

終於要開始建立發揮 ChatGPT 功能的程式。以下將思考 RPG 遊戲的戰鬥場景，製作由 ChatGPT 生成角色台詞的程式。

1 思考作品

在 RPG 遊戲的戰鬥場景中，主角與敵人雙方的 HP（生命值，也就是剩餘的能量，歸零代表失敗）會隨著攻擊而不斷變化。如果想讓角色人物根據 HP 的變化說出各種台詞時，一般程式可能需要撰寫大量的條件式。

不過，使用 ChatGPT 只要提供雙方的 HP 狀態，就能自動生成適合的台詞。因此，以下將思考如何設定簡單的 RPG 遊戲。

登場角色

- 主角（騎士）
- 敵人（飛龍）

參數

- 主角的 HP
- 敵人的 HP

條件

- HP 的預設值皆為 100
- HP 降至 0 以下即失敗

希望 ChatGPT 執行的工作

- 根據主角和敵人的 HP 生成敵人的台詞

操作方法

這裡為了學習將 ChatGPT 整合到程式的方法而盡量簡化遊戲的操作機制。

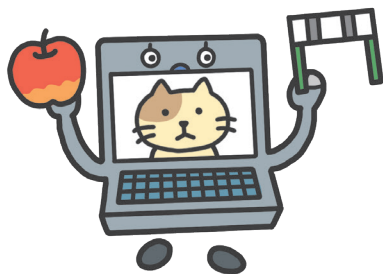
每次按下空白鍵時，主角或敵人就會發動攻擊。隨機決定由誰進行攻擊，造成的傷害值也是透過亂數隨機決定。

6

進階篇

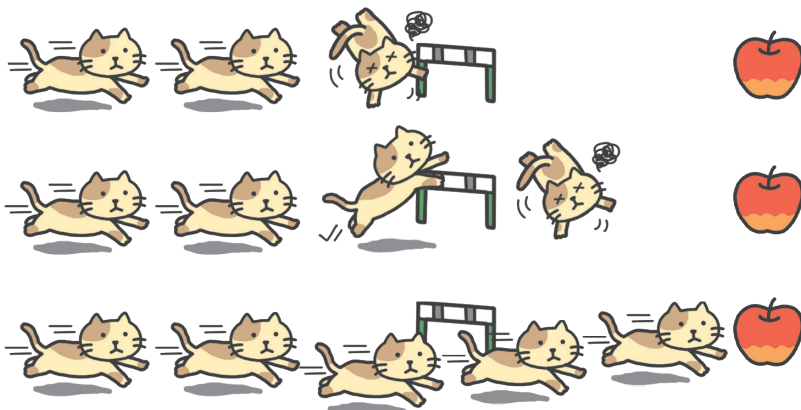
用遺傳演算法 讓貓咪的動作進化

在前面製作的程式中，機器學習都是由 Scratch 的擴充功能負責處理，但是這一章要使用 Scratch 設計演算法。所謂的機器學習並非是解決某個問題時，由人類組合所有程式並下達指令，而是機器，也就是電腦自行學習，導出解決方法。這裡要介紹的「遺傳演算法」也是一種機器學習，本章的內容比較難，當你看完前面四章，想進一步了解機器學習的話，請務必挑戰看看。



我們要讓貓咪不斷世代交替，
變得更聰明。剛開始無法順利
抵達蘋果的貓咪會愈來愈接近
蘋果喔！

OK！開始，跳！



效率真的愈來愈好了！



小孩遺傳了父母的基因，
會學習如何才能成功又快速地抵達目標。
這個過程會重複好幾代。

我們居然可以用程式設計做到像神一樣的事情？



雖然難度比前面高，但是你們一定沒問題的。



我會努力的！



6 章

進階篇
—— 用遺傳演算法讓貓咪的動作進化